AFRL-IF-RS-TR-2003-260
Final Technical Report
November 2003

ANALYTIC PREDICTION OF EMERGENT DYNAMICS FOR AUTONOMOUS NEGOTIATING TEAM (ANT) SYSTEMS

Utah State University

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. K279

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-260 has been reviewed and is approved for publication

APPROVED: /s/
ROBERT J. PARAGI
Project Engineer

FOR THE DIRECTOR: /s/
JAMES A. COLLINS, Acting Chief
Information Technology Division
Information Directorate

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 074-0188*

1. AGENCY USE ONLY (Leave blank)	**2. REPORT DATE**	**3. REPORT TYPE AND DATES COVERED**
	Nov 03	Final May 00 – May 03

4. TITLE AND SUBTITLE

ANALYTIC PREDICTION OF EMERGENT DYNAMICS FOR AUTONOMOUS NEGOTIATING TEAM (ANT) SYSTEMS

5. FUNDING NUMBERS

C - F30602-00-2-0532
PE - 62301E
PR - ANTS
TA - 00
WU - 05

6. AUTHOR(S)

Daniel Watson, Todd Moon and James Powell

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Utah State University
Contracts and Grants Office
1415 Old Main Hill
Logan UT 94322-1414

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/IFTB
525 Brooks Rd
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2003-260

11. SUPPLEMENTARY NOTES

AFRL Project Engineer: Robert Paragi, IFTB, 315-330-3547, paragir@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***

In the work on analytic prediction and dynamics at USU, three distinct perspectives have been brought to bear. The computer science perspective was closely tied to real-time scheduling and planning questions, leading to case-based negotiation. In this work, autonomous negotiating systems are composed of logically separated software agents that control resources that altruistically seek to perform useful work in a cooperative manner. This study examines the negotiation strategy that improves over time by gained experience. A case-based negotiation strategy is presented that allows self-organized scheduling of the tasks.
The mathematics perspective examines task completion under a general resource allocation model, where the allocation problem as a nonlinear differential equation, which was used to predict completion ability. This predictive model was then compared with simulation models.
The electrical engineering perspective examined the praxeic decision theory approach to multiple agent coordination. Inference, the problem of estimating the goals of other agents in the arena, is discussed in the praxiec context. Another viewpoint toward multiple agents systems is also presented using catastrophe theory. In this analysis it is determined that a “phase transition” behavior is to be expected.

14. SUBJECT TERMS

autonomous negotiation, autonomous agents, software agent system complexity and dynamics, intelligent agents

15. NUMBER OF PAGES

140

16. PRICE CODE

SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Contents

List of Figures

Chapter 1

Introduction

The work on analytic prediction of emergent dynamics undertaken at Utah State University has been itself an exercise in multiagent coordination. Three distinct perspectives, drawn from the Computer Science Department, the Mathematics Department, and the Electrical and Computer Engineering Department, were brought to bear on problems of multagent coordination.

- The computer science perspective was closely tied to real-time scheduling and planning questions, leading to case-based negotiation. In this work, described in chapter 2, autonomous negotiating systems are composed of logically separated software agents that control resources that altruistically seek to perform useful work in a cooperative manner. The work environment is classified into resources, tasks and missions. Each resource has a predefined set of functionalities that define the actions that the resources can perform, and each task requires one or more functionalities to be applied to it for a specific amount of time. All resources providing the requisite functionalities must rendezvous for the duration of that time in order to complete task. Each mission is composed of a set of tasks and a partial ordering among those tasks represented with a directed acyclic graph. Missions, tasks, and resources are represented by software agents. This study examines the negotiation strategy between those agents, using a negotiation strategy that improves over time by gained experience. A case-based negotiation strategy is presented that allows self-organized scheduling of the tasks. Through software simulations, the study shows that important characteristics of system performance are positively affected by such experience-based negotiations.

- The mathematics department examined questions of task completion under a general resource allocation model, as discussed in chapter 3. The allocation problem was examined as a nonlinear differential equation, which was used to predict completion ability. This predictive model was then compared with simulation models. An appendix describes the simulation software.

- The electrical engineering department examined the "praxeic decision theory" approach to multiple agent coordination. Chapter 4 introduces the concept, beginning with single agent systems and then extending to multiple agent systems in negotiation. Inference — the problem of estimating the goals of other agents in the arena — is also discussed. Another viewpoint toward multiple agent systems is also presented using catastrophe theory. Two distinct nonlinear models for multiagent behavior are examined. In both cases, it is determined that a “phase transition” behavior is to be expected. This phase transition behavior is distinct from the type of phase transitions from “easy problems” to “hard problems” frequently discussed in the multi-agent literature, which is due to systems with large number of agents. This has to do with the nonlinearity, which gives rise to “cusp” singularities on the manifold of parameter spaces.

Chapter 2

Organizing Missions for Autonomous Resources Using Case-Based Negotiation

2.1 Introduction

Autonomous Negotiating Systems are composed of logically (even geographically) separated software agents that control logical or physical resources that altruistically seek to perform useful work in a cooperative manner.

This study examines the negotiation strategy between autonomous agents, using a negotiation strategy that improves over time by gained experience. A case-based negotiation strategy is presented that allows self-organized scheduling of tasks on distributed resources. Through software simulations, this study shows that important characteristics of system performance are positively affected by such experience-based negotiations.

It is often useful to classify the work environment into resources, tasks and missions. Generally speaking, tasks represent work to be accomplished, resources represent items used to achieve work, and missions represent overall goals that can be accomplished by the successful completion of one or more tasks.

Each resource has a predefined set of functionalities that define the actions that resources can perform. Resources can perform at most one functionality at a time, and may need a startup time, t_{startup}, (as in the case of travel time for physically distributed resources,) before the appropriate functionality can be applied. In this study, t_{startup} is assumed to be 0.

Each task requires one or more functionalities to be applied to it for a specific amount of time. It is assumed that all resources providing the requisite functionalities must rendezvous for the duration of that time in order to complete the task. Tasks are also ascribed an arrival time, t_{arrival}, indicating the time a task enters the system and a need for work to be accomplished, an earliest start time, t_{earliest}, before which no work can be performed, and a deadline, t_{deadline}, before which all work must be completed. Tasks that are not completed before their deadline fail, and are summarily removed from the system.

Each mission is composed of a set of tasks, a partial ordering among those tasks, t_{arrival}, t_{earliest}, and t_{deadline}.

This partial ordering of tasks can be represented with a directed acyclic graph (DAG) where the nodes of the graph represent tasks and each arc (*i*, *j*) from node *i* to node *j* represents a time-ordering of *i* and *j* (i.e., task *i* must be completed before task *j* can begin).

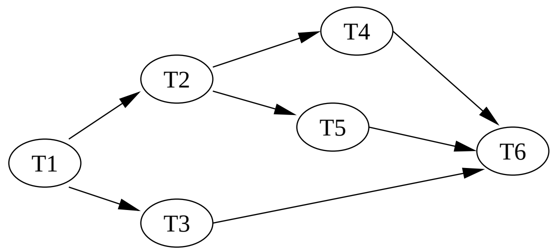


Figure 2.1: Example of partially ordered tasks in a mission

As an example, for the mission represented in Figure 1, task 1 (T1) must be completed before T2 or T3 can begin, T2 must be completed before T4 or T5 can begin, and all the other tasks must be completed before T6 can begin.

Each resource is controlled by a resource agent that is responsible for finding useful work for that resource to perform. The resource agent negotiates with other agents to arrive at a schedule of work for that resource, maintains that resource's schedule of work to perform, and directs the resource when to begin and end work for each task it is scheduled to participate in. In this study there is a one-to-one correlation between resources and resource agents, although in general one resource agent might be responsible for many resources.

In the autonomous system, there is a community of task agents. Each task agent is responsible for overseeing the completion of tasks, including finding and booting resources for that task and monitoring the progress of those tasks. Task agents have permanence in that they oversee the completion of many tasks during their life time. In this study, each task agent supervises at most one task at a time, from the time the work is first requested until the time the work is completed (or the task fails). New task agents are created as needed; thus the number of task agents in the system is equal to the maximum number of known tasks at any single moment in time. The removal of task agents from the system is not considered here.

In the same manner, each mission is supervised by a mission agent that is responsible for finding a task agent for each of its member tasks. Mission agents are also given the responsibility for determining the t_{arrival}, t_{earliest} and t_{deadline} for each task, based on the t_{arrival}, t_{earliest} and t_{deadline} for the overall mission.

Mission, task, and resource agents negotiate to determine acceptable allocations of resources to tasks extended in time. The negotiation strategies are founded on case-based negotiation.

In case-based negotiation [1], a case stores successful and unsuccessful negotiating strategies gained from experience. Mission and task, each agent maintains a library of cases that are created and refined as that agent negotiates with others.

Results of a simulation study in which the case-based negotiation is compared to a simple strategy that does not rely on experience indicate that there is a positive effect of experiential learning on the negotiation process, and that experience-based autonomous scheduling strategies can adapt to new environments without intervention.

The rest of the paper is organized as follows: Section 2 introduces current related works by other investigators in this field, most specifically those examining case-based strategies. Section 3 focuses on the details of negotiation strategy, including the negotiating mechanism, case definition, case parameters, and algorithms. The different simulation experiments are described in Section 4. Results from a set of different approaches are presented and analysis for each result is given. Summarizing and concluding remarks are provided in Section 5.

2.2 Background

Research into the behavior and uses of software agents is varied and widespread. A unifying theme is in examining the potential for software agents to exhibit expertise through competition or cooperation. Agents may be managing private resources as in the case of web agents [2] and email highlighting agents [3], among others. Some systems employ multiple agents [4] that adapt to the current community of agents, while other systems rely on single agents [5] that 'travel' in a distributed environment, adapting to diverse conditions and providing functionality that would otherwise be cumbersome, perhaps even infeasible.

Agents can negotiate using different models, such as declarative descriptions [6] that rely on rule-based representation language to automate negotiations of business contracts, commitments [7], that capture the obligations from one party to another, and argumentative negotiation [8], which is based on values of private information and preferences.

Negotiation between agents can occur in a single transaction, or can be accomplished in several steps, as in [9], which introduces a multidimensional, multi-step negotiation mechanism for task allocations among agents.

As in this study, the multi-step negotiating strategy improves over time, while [9] improves by constructing multiple protocols that adapt to different situations.

Resource allocation can be determined by applying schema globally instead of negotiating. [10] presents the Marbles schemes, a family of cooperative and adaptive algorithms in which all the requirements and resource properties are known a priori.

Negotiation efficiency can be improved by calculating statistics on interaction performance. [2] discusses the efficiency improvement for interactions of WebAgent. By knowing the distribution of access time, an agent can optimize the access strategy (or negotiation strategy). Thus, statistical results can be applied to agent negotiations to improve the performance. Whenever the access is not stable (e.g., the internet connection is interrupted), this strategy is useful in determining when it is appropriate to renew access to the previous site, or to a new site.

Case-based negotiating [1] can be applied to resource-private agent systems, and is a good example of how an agent negotiates for the use of other resources in order to complete one or more tasks promptly. Negotiations use case-based reasoning [11] to learn, select, and apply negotiation strategies. Case-based reasoning is used as a basis for this study and is more fully described in Section 3.

This paper differs from other papers in focusing on the negotiation between task agents and resources. The contribution of this paper is in providing a case-based negotiation strategy between task agents and resources to achieve a solution for mission completion, specifically in scenarios where tasks are scheduled for resources in advance due to the known ordering of tasks inside a mission. This paper shows that case-based negotiation can be beneficial. A defining element that distinguishes this study from others [12, 13, 7, 6] is the juxtaposition of autonomy, deadlines, and the focus on systems that are loaded to the point of task failure as a result of missed deadlines.

2.3 Case-based Negotiating

An argumentative negotiation is adopted in this study where task agents negotiate with resources by presenting one or more arguments to convince resource agents to allocate their resources to the task. An argument is an expression indicating the value of a feature of an agent, in a form of: ¡feature¿ ¡comparison operator¿ ¡value¿ (e.g., priority = high, negotiation time = 10, etc.).

Several primitives are defined as negotiation messages (e.g., 'require', 'accept', 'decline', etc.), each of which has its own parameters. In general, task agents request resource agents to fulfill their functionalities by supplying a list of arguments in order to convince the resource agents. The resource agent then evaluates those arguments and replies to the task agent acceptance or rejection of the task agent's request. A resource agent may make a counteroffer in the form of its own argument. The task agent may then accept, decline or make a counteroffer again until both sides make an agreement.

Case-based negotiation is an application of case-based reasoning (CBR) [11]. Instead of giving a diagnosis or solution to a problem, this study uses the diagnosis or solution as the current negotiation strategy. [1] presents a case-based negotiation. [1] uses CBR to select, apply, and learn the negotiation strategies that the agent uses.

In [1], the negotiation of agents is targeted at requesting resources from other cooperative agents. Each resource is local to an individual agent. An agent must negotiate with others in the cooperation of resource uses to complete tasks. An agent uses different negotiating strategies at different instants, because the current agent status or the global environment variables differ dynamically. Each agent stores those different negotiation strategies by cases, which store the negotiation parameters (or strategies) under certain of environment.

This study is based on distributed (or shared) resources. It is targeted at presenting a solution on resource scheduling and allocation in a multitask, multi-resource and soft real-time environment, where the partial ordering of tasks are known and any task may fail due to the lack of competition. The structure of each mission (i.e., tasks and needed functionalities) is assumed to be known in advance. The dynamic creation or redefinition is not considered here. This study does not examine the real-time case where the negotiation is limited by time, although the focus here is in finding allocations of resources quickly.

2.3.1 Case Library

Each task agent maintains a case library. Each case retains environment descriptors and negotiation parameters from previous negotiation transactions that may be used in the current negotiation if current environment is closely related to the case environment.

Each case is composed of two parts. The first part is a vector of descriptors of that case's environment. Each descriptor describes the value of a feature of the case's environment. For example, negotiation time for current task. The second part of the case is the set of negotiating parameters, such as priority, number of unfilled functionalities, etc.

In this study, the environment descriptor portion of a case is composed of three parts: system information, self information, and resource information.

System information includes the ratio of the number of required functionalities by active tasks versus the number of supplied functionalities by available resources. This parameter is based on data that is perceived by each individual agent, gained by monitoring requests and responses by task and resource agents over the communication medium. This is useful for determining how competitive it is for current tasks to request resources. Additionally, system information includes the ratio of average negotiation time for active tasks to the maximum negotiation time perceived by that agent. This parameter is intended to measure to what extent active tasks can alleviate the load of system.

Self information of the task agent is composed of two elements. The first element is the ratio of self-negotiation time to the maximum negotiation time, as a measure of past negotiating performance; and the second element is the ratio of the number of required functionalities of this task to the possible maximum number of functionalities known to be required by any task. This variable measures the relative difficulty of the task to fulfill its functionalities.

Resource information for this study adopts one variable: the ratio of the number of free resources to the total number of resources perceived by any agent under the rationale that, if more resources exist, tasks are easier to be fulfilled.

The set of negotiating parameters for each case in this study is composed of four elements: priority, time required, the number of unfilled functionalities and the ratio of $t_{earliest} - t_{arrival}$ to $t_{deadline} - t_{arrival}$, which indicates the percentage of total time before the deadline that can be used for negotiation. The priority, time required, and the number of unfilled functions for each task are nonnegotiable. The negotiation time between tasks and resources is adjustable by each task. Tasks in general seek a low value for negotiation time, because tasks desire to complete as soon as possible.

2.3.2 Case Selection and Retrieval

The case-based negotiating strategy in general evaluates cases using weighted matching, and employing different matching functions for different features. For example, the environment descriptors in Figure 2 have two features: A and B. Each feature is assigned a similarity function $\text{Similarity}_i\,(i, j)$ that calculates the similarity of two values of this feature. The overall similarity of any two environments is calculated by weighted sum of each feature similarities. After evaluation, the most similar case (i.e., the one with maximum similarity result) will be selected.

After the most similar case is selected, the strategy from this case is used to control the negotiation.

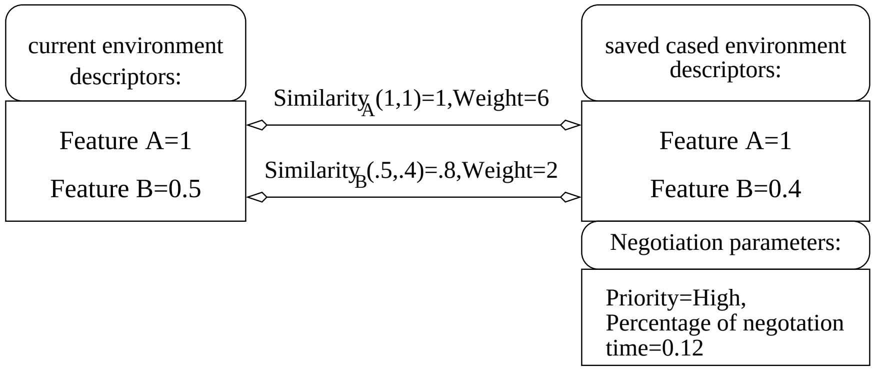


Figure 2.2: An example of similarity comparison

2.3.3 Case Storage and Learning

If the negotiation fails, the task agent refines the negotiation strategy in a ‘conservative’ way (e.g., as in this study, increasing the percentage of time before deadline that is allocated to negotiation). Conversely, if the negotiation succeeds, the task agent refines the negotiation strategy in a more ‘adventurous’ way (e.g., as in this study, decreasing the percentage of time allocated to negotiation). After the refinement, the new case is stored in the case library.

Before storing a new case, it is compared to the most similar case that already exists in the case library. If they are similar within a threshold, the new case is discarded so that the case library remains a stable size. The strategy in the new case will compromise with that in the similar case. For example, the compromise can be taken by averaging the two values of negotiating parameters. Some new cases are identified as ‘irrational’, and are discarded anyway, such as a case with more insufficient resources environment, ask for less negotiating time, comparing to the strategy in the similar case.

2.3.4 Task Algorithm

Each task agent negotiates with available resources until all of its functionalities have been filled. A task can find all the possible resources by collecting responses from available resources after it broadcasts a ‘request’ message to all resources. After the task chooses a resource and sends out the request to this resource, it may get an ‘accept’ or a ‘decline’ message from the resource. The following is a formalized algorithm on task side:

L_1. If all the functionalities have been filled, go to L_6. Otherwise, get the next unfilled functionality,

L_2. Broadcast the request for the current functionality, and set up a set of resource agents as potential negotiators by monitoring responses from resource agents,

L_3. Choose next resource agent from the set of resource agents, determine the current environment, seek the case library, and try to find the matching case,

If the matching case is not found, create a new case with default arguments. Prepare negotiation with the current resource agent,

If the matching case is found, fetch the arguments from the case. Prepare negotiation with the current resource agent,

L_4. Send a request to the current resource agent,

L_5. Wait until one of followings happens:

If the task runs out of negotiation time, release all the resources previously scheduled, refine the negotiation strategy, and store this case into the case library. Go to L_7,

If the task receives an 'accept' message, put this resource into a scheduled resource list, and go to L_1,

If the task receives a 'decline' message, go to L_3 if the counteroffer is impossible, or go to L_4 if the counteroffer is adopted,

L_6. The task starts executing. After it finishes, refine the negotiation strategy, store this new case into the case library and release all resources previously occupied,

L_7. Save the statistical data and exit.

2.3.5 Resource Algorithm

Each resource has 3 states: idle, scheduled and active. Scheduled resources can be grabbed (i.e., allocated to) by other higher priority tasks, but active resources cannot be grabbed by any task, idle resources can of course be grabbed by any task. Upon a request from a task, resources evaluate the arguments passed by the negotiation message, and make a decision based on the result from the utility function that is used to evaluate the importance of a task. The utility function of each resource maintains a threshold, which measures how strong (or how important) the arguments are. Decision is made by comparison between the result from the utility function and the current threshold. A formalized algorithm on the resource side is as follows:

L_1. Wait until one of followings happens:

If a 'request' message is received from any task, go to L_2,

If an 'activate' message is received, change current state to 'active'. Go to L_1,

If a 'release' message is received, go to L_4,

L_2. Calculate the result from the utility function parameterized by the arguments from the negotiation,

L_3. Identify current state,

If current state is 'idle', send 'accept' message back, change the state to 'scheduled', set up a new threshold by the result, and remember the scheduled task and its functionality. Go to L_1,

If current state is 'scheduled', task competes for the resource. If the result is higher than the current threshold, the resource will send 'accept' message back while informing the already scheduled task of 'lost resources'. Change currently scheduled task to this new task and renew the functionality. Go to L_1; otherwise, send 'decline' message back, and go to L_1,

If current state is 'active', send 'decline' message back with a counteroffer, which indicates the left time the resource keeps 'active'. Go to L_1,

L_4. Change current threshold to 0, mark currently scheduled task and functionality as null, and reset current state as 'idle'. Go to L_1.

2.4 Experiments

Figure 3 shows the experimental model. A task distributor generates and distributes new tasks, simulating the pattern of partially-ordered tasks in the missions. Task agents accept tasks from the task distributor. Each task agent accepts at most one task at a time. Each task agent dispatches a thread for each task and negotiates with resources to fulfill all functionalities required by this task. Task agents are responsible for the maintenance of their case libraries, including new case insertion, old case refinement, and case removal. Negotiation occurs between task agents and resource agents (e.g., in Figure 3, there are 20 resource (or resource agents) available). The result of each task is recorded after it completes (or fails).

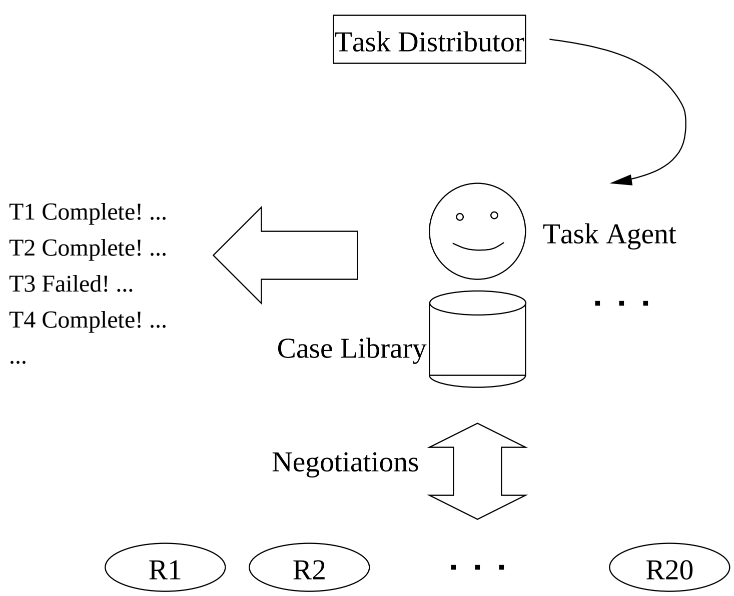


Figure 2.3: Experimental Model

Before conducting the experiments, some predetermined values for some experimental parameters are required (e.g., the number of tasks, number of resources etc.). The following predetermined parameters remain unchanged in this study.

Number of resources=20, a parameter describes how many resources are available in the system.

Maximum number of functionalities=8, a parameter indicating the maximum number of different functionalities that can be set up inside a task or a resource.

Maximum number of functionalities per task or resource=3. This parameter is used to set an upper bound for the number of functionalities that can be required by any task or be offered by any resource.

Levels of priority=3. Priority is a parameter used in the case-based negotiation. Each task has a predetermined value for its priority (e.g., ‘high’, ‘medium’ or ‘low’).

Maximum negotiation time=20 (time units). This parameter is adjustable by each task, and resource can make counteroffer to a task on this parameter.

Maximum running time=20. This parameter indicates the maximum time the task can run after all the functionalities allocated to this task are ‘active’ on it.

Task sample collection rate=10. This task calculation interval indicates the number of tasks that complete before the next graph point is calculated.

Based on the above parameters, a simple program is used to produce a random set of resources and tasks so that the internal functionalities offered by each resource and required by each task are randomly distributed.

Experiments are conducted to observe (1) average task completion time under case-based negotiation with varying task arrival rates at different times. The goal is to examine the impact that cases put on the task completion over time. With a variance on task arrival rate, another group of cases adapting to the changing environment are expected. The experiments also observe the learning rate of cases under different refinement strategies; (2) average task completion time under simple negotiation strategy with different percentage of negotiation time that remains constant in simple negotiation. Experiments are expected to examine the performance on task completion under different percentage of negotiation time; (3) average completion time with different case granularity in case-based negotiation.

By changing task-distributing intensity, the volume of task stream (or workload of system) can be adjusted.

(1) Figure 4 shows an average task completion time under case-based negotiation with different refinement strategies. There are two refinement strategies in these experiments: one is an ‘aggressive’ refinement strategy, the other is a ‘conservative’ refinement strategy. The ‘aggressive’ refinement strategy increases the percentage of negotiation time by 0.2 if each task fails, and keeps the original percentage of negotiation time if each task completes. The ‘conservative’ refinement strategy increases the percentage of negotiation time by 0.05 if each task fails, and decreases the percentage of negotiation time by 0.01 if each task completes. As shown in Figure 4, task arrival rate changes from 0.2 per time unit to 1 per time unit at the 100^{th} time cycle.

Under each of refinement strategies, there are initially no cases in the case library. Because the negotiation time is initially a small value, there is a significant possibility for each task to fail at the starting phase of negotiation. As a result, the average completion time is low at the beginning. After some time, cases become adapted to the current environment and negotiation time (i.e., the amount of time in which agents are allowed to negotiate before resources become active on that task) increases. Average task completion time increases as a result of that more tasks complete (or fewer tasks fail). After the task arrival rate changes from 0.2 per time units to 1 per time unit (at the

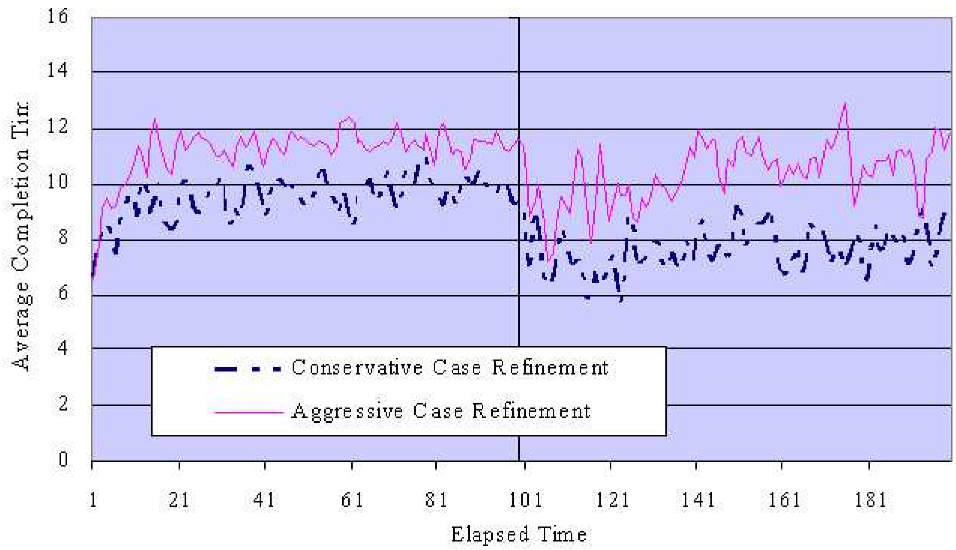


Figure 2.4: Average task completion time in case-based negotiation

100th time cycle), there is a significant drop in average task completion time because there are no such cases suitable for the current new environment. Agents begin making 'conservative' decisions by limiting negotiation time to provide a 'safety net' of additional time in which to complete a task. After some time, new cases are set up that are adapted to new environments, and the average task completion time improves.

Comparing two different refinement strategies, Figure 4 demonstrates that the 'aggressive' refinement strategy brings a faster learning rate of cases (or the rate of adapting to new environments) than the 'conservative' refinement strategy.

(2) Under simple negotiation, the percentage of negotiation time is constant, and cannot be changed during the negotiation. Figure 5 demonstrates average task completion time in different percentage of negotiation time. Task arrival rate changes from 0.2 per time unit to 1 per time unit at 100th time cycle.

Figure 5 indicates that, in a simple negotiation, average task completion time keeps in a constant range unless the task arrival rate changes. After task arrival rate changes from a 'slow' arrival rate to a 'fast' arrival rate (i.e., changes at 100th time cycle in Figure 5), the average task completion time increases only if the percentage of negotiation time (percentage of negotiation time = 0.5 as in Figure 5) still accommodates current heavier workload. Conversely, the average task completion time decreases if current negotiation time (percentage of negotiation time = 0.33 or 0.25 in Figure 5) cannot accommodate current workload. Based on these phenomena, a constant percentage of negotiation time has a possibility to fail due to lack of adapting to varied environments.

Case-based negotiation is able to adjust negotiation parameters to adapt to new environments. Therefore, case-based negotiation shows a positive effect comparing to the simple negotiation.

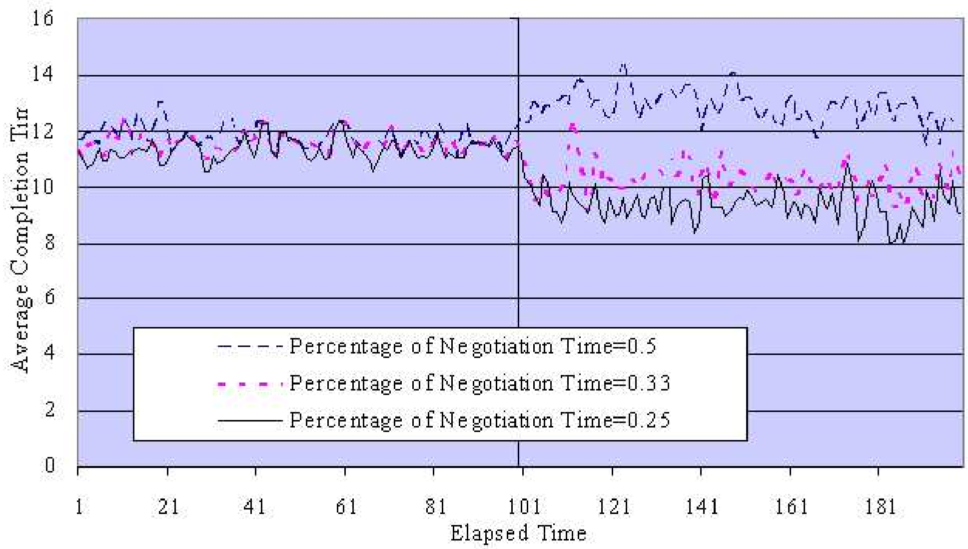


Figure 2.5: Average task completion time in simple negotiation

Comparing the average task completion time in the simple negotiation, case-based negotiation can also hit a high task completion time by speeding up the case-learning rate.

(3) By changing the threshold of similarity function that determines the difference of cases, case library can have different case granularity. Figure 6 indicates that different granularity causes different learning speed of new cases so that it will takes longer or sooner for the cases to adapt to new environments. As shown in Figure 6, in a low task arrival rate, granularity doesn't significantly affect the completion time, because each case is not sensitive under a low workload environment. Conversely, in a high task arrival rate environment, the average task completion time with low granularity cases is higher than that with high granularity cases.

2.5 Summary

As a characteristic of negotiation performance, task average completion time has been prolonged after cases have been learned. The jagged curve shows a gradually stabilized completion time accompanying with the learning of new cases or the refining of old cases. The simple negotiation keeps a relatively constant completion time, instead of showing an improved or gradually stabilized curve. These results also demonstrate that, whenever an extreme change happens in the system, new cases are created and another round of case learning will initiate and become stabilized after a period of time in case-based negotiation.

Results from different case refinement strategy and granularity on case learning indicate that the different rate of stabilization that occurs in average task completion time. Aggressive refinement

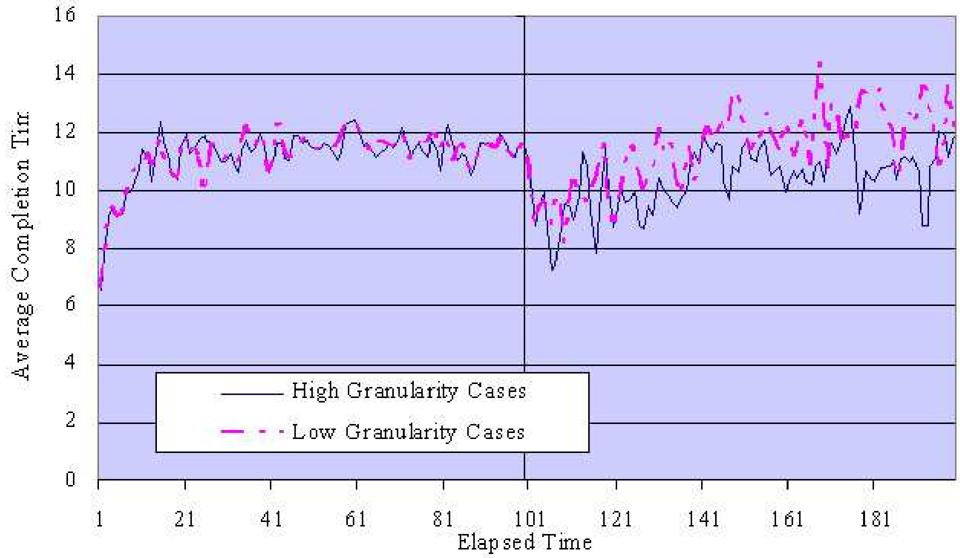


Figure 2.6: Average task completion time in different granularity cases

strategy and low granularity cases make the average completion time become stabilized faster than conservative refinement strategy and high granularity cases, because aggressive refinement strategies make cases adapt new environments faster, and low granularity alleviate the sensitivity of environmental changes.

Case-based negotiation shows a significant benefit on adapting different environments. But in the simple negotiation, constant negotiation parameters take risks to bring a low task completion time due to inability in adapting to new environments. Case-based negotiation is able to bring a high task completion time by speeding up adapting rate comparing to the benefits some 'generous' parameters create in the simple negotiation.

Chapter 3

Agent-based Task Completion

3.1 Predicting Agent-based Task Completion

3.1.1 Summary of Results

This chapter presents a model for solving a resource allocation problem (the ‘Screaming Generals’ problem) in which autonomous agents negotiate for use of the resources. The Screaming Generals problem is a test-bed for our ideas about task completion in a multi-agent environment with hard deadlines. Rather than analyze some specific negotiation scheme, we present a model that accepts the *results* of negotiation as input. Our characterization of the results of negotiation is based on the priorities built in to the negotiation scheme – for example, some negotiations commonly result in an approximately equal distribution of resources. Three different negotiation strategies are presented, and while these are by no means exhaustive, our framework easily accommodates the addition of new strategies. We then analyze how well the agents complete tasks under different negotiation inputs, using both analytical and numerical techniques. Our numerical techniques allow us to determine the regimes in which given negotiation strategies are superior to others, and to estimate the asymptotic rates of task completion as the number of tasks increases.

3.1.2 Introduction

In the context of computer science, an agent is some entity (whether virtual or physical) that has control over its own actions. A variety of applications have been found for agents, some involving searching or bidding over networks (such as the internet). The Defense Advanced Research Projects Administration (hereafter DARPA) is interested in using them to replace conventional human and computer resources in applications such as logistics, reconnaissance, and combat. Agents have the advantage of being able to make decisions on their own while still being able to communicate with other agents. However, if the physical housing of the agent is damaged or destroyed, other agents are not dependent on the missing agent, and no lives are lost. Our research mandate from DARPA was to begin analyzing systems of agents to observe how they perform.

Consider a set of tasks which require the use of some set of resources for their completion or performance. Some examples include assigning CPU cycles in a Beowulf cluster, radar emitters in a naval fleet, or more basically slots in a schedule. A classical approach would be to decide on a distribution of resources that would allow the tasks (or some portion of them) to be completed. While this method has its merits and has been widely studied (*e.g.* scheduling, linear programming), it is a centralized approach – a unique solution is determined and resources are allocated accordingly.

The centralized method also presupposes that the entity determining the solution has control of the resources as well as responsibility for scheduling appropriate allocations of those resources. For our purposes an agent is similar to this entity, except that it only controls some subset (possibly empty) of the resources. A collection of these agents forms an autonomous system, in which the agents can negotiate with one another for use of the resources. Completion of tasks depends on the behavior of the agents. If the negotiation occurs in a time-critical environment it introduces an interesting trade-off between negotiation and task completion. Even under the assumption that an agent can ‘talk’ and ‘work’ at the same time, time spent negotiating can still produce a delay in reallocating the resources to adapt to a change in circumstances.

This is a general description of the agent-based approach which could be adapted to a wide variety of problems. For purposes of this paper we will discuss a narrower regime in which the details of the negotiation are suppressed. Regardless of how the agents actually conduct negotiations, they will arrange for some distribution of resources (presumably in finite time). As an example, consider a simple bazaar system. Initially agents are assigned a certain amount of money, which may depend on the importance of their task, its degree of completion, and its proximity to deadline. The agents then bid on a large supply of homogeneous widgets. Agents controlling widgets make counter-offers, and in general a price is agreed upon (after negotiation) that is somewhere in the middle. After all the money is spent, each bidder will have a number of widgets proportional to the amount of money it was given initially. We introduce the concept of resource allocation strategies to describe this end-result. Thus we only consider two aspects which result from negotiation – the final allocation and the time spent reaching that allocation. Perhaps the agents have the goal of hammering out a fair share of the resources for each agent. Conceivably there are many ways to do this, but as far as the completion of tasks is concerned all that matters is how long it takes to achieve the fair division. Any other goal, such as completing smaller tasks or critical tasks first, can be accommodated by these strategies. Our goal in this paper is to develop a modeling philosophy for describing task completion by autonomous agents and determine the conditions under which a given strategy is superior to other proposed strategies.

We have considered some scenarios that could be analyzed in this manner. The first is completing tasks in a distributed computing environment (*e.g.* a Beowulf cluster). Tasks can be assigned processing time according to the size of the task, the task’s deadline, the task’s assigned priority, or other factors. The tasks can be any problem that can be usefully split into pieces such as list sorting or signal processing. Each task has an agent assigned to complete it by negotiating with the other agents for use of CPU time.

Another example is the DARPA–ANTS challenge problem. ANTS stands for Autonomous Negotiating TeamS. This problem is interesting precisely because of the possibility of a decentralized solution. A decentralized network presents no obvious or critical target for an enemy to focus on, and can presumably function just as well if a few nodes are destroyed. In the challenge problem several radar sensors are positioned around a model railroad and tasked with tracking one or more trains. Monitoring a radar ‘track’ for each train is a task with a responsible agent and the radar stations and the timing of their emissions are the resources, controlled by other agents. In this problem, as in many others, central questions are: under what conditions can successful task completion be guaranteed, and how does negotiation overhead influence task completion?

In this paper, we will first present a conceptual model called the ‘Screaming Generals’ problem, which will allow us to address these questions. This formulation is independent of any specific application. After describing this problem in the form of a system of differential equations, we propose three resource allocation strategies and proceed to analyze solution characteristics . We then show the results of a numerical simulation of the problem, using the different strategies. Our results will illustrate two points: first, that our analytical methods provide insights into the nature and complexity of the problem, and that we can bound the performance of a resource allocation strategy. Second, by using numerical simulation and data fitting we can determine the best strategy for given conditions.

3.2 Modeling Task Completion

3.2.1 The Screaming Generals Problem

We are specifically considering *divisible* tasks, that is, tasks whose accomplishment can theoretically be subdivided into many small (ideally identical) portions. In our Beowulf cluster example, this is true for list sorting, image processing, and numerical computations, among other things (these tasks do not necessarily have to be done in parallel). For each task this provides a natural index of completion: the fraction of the task which remains undone (F_j). By examining how this fraction decreases in time we will be able to predict how different strategies for resource allocation impact the completion of individual tasks. Furthermore the tasks are time-sensitive in that they must be completed by a certain deadline or else be considered total failures. Deadlines are critical because many problems need to be solved in some finite amount of real time. The radar tracking problem, for instance, has very definite deadlines based on the hardware requirements and the demands of physics – if the sensors spend too much time negotiating they will not have enough time to produce accurate tracking results before the target moves on.

As a conceptual model for divisible tasks we think of ditches. Each ditch, labeled j, requires a certain number of man-hours, R_j, to dig. Each ditch has a general who has overall responsibility for making sure that the ditch gets dug, and who negotiates for men with other generals, all from a fixed pool of M men on base. A basic model for resource allocation is by how loudly each general ‘yells’ in comparison to the other generals. Based on a variety of factors (proximity of deadline, length of ditch, etc.) a general may choose to negotiate at greater or lesser volume. The number of men a general receives on an hourly basis is in direct proportion to the volume at which the general is yelling. By building various models for how a general’s loudness varies with ditch completeness and deadline we will examine how different negotiating outcomes affect the rate of task completion.

Quantity	Units	Description
j	–	Task (ditch index)
N	–	Number of currently active tasks
F_j	–	Fraction of ditch j remaining un-dug
M	men	Total number of men available to dig all ditches
f_j	–	Fraction of total resources currently allocated to task j
t	hrs	Current time
t_j	hrs	Time when task j began
D_j	hrs	Deadline for completion of task j
L_j	feet	Total length of ditch j
s_j	feet	Distance currently dug along ditch j
$r_j(s_j)$	man-hrs/feet	Work density required to dig ditch at a distance s_j along the ditch
R_j	man-hrs	Total number of man hours required to complete task j

Figure 3.1: List of parameters and variables used.

3.2.2 Task Completion Modeling

The fraction of ditch j remaining to be dug, F_j, is given by

$$F_j = \frac{L_j - s_j}{L_j} = 1 - \frac{s_j}{L_j}, \tag{3.1}$$

where s_j is the distance currently dug and L_j is the length of the ditch. In the case of a ditch with variable consistency (and therefore varying difficulty in digging along its length) a input-output constitutive relation holds for progress:
Man-hours required to dig a distance Δs at a spot s_j feet into the ditch= $r_j(s_j)\Delta s$ = $f_j M \Delta t$ = *Man-hours allocated to task j for the amount of time Δt required to dig a distance Δs.*

Here $r_j(s_j)$ is the work density required at distance s_j along the ditch and f_j is the fraction of men M assigned to task j as a result of negotiation. Thus, progress along the ditch obeys the relationship

$$\lim_{\Delta t \to 0} \frac{\Delta s}{\Delta t} = \frac{M f_j}{r_j(s_j)} \Rightarrow \frac{ds_j}{dt} = \frac{M f_j}{r_j(s_j)}. \tag{3.2}$$

Differentiating (3.1) gives

$$\frac{d}{dt} F_j = -\frac{1}{L_j}\frac{ds_j}{dt} = -\frac{M f_j}{L_j r_j(s_j)}. \tag{3.3}$$

In the case of a homogeneous ditch (one which requires equal resource per distance), the general can estimate the total resource commitment required to dig the ditch, R_j, as $R_j = r_j L_j$, where r_j is constant with distance along the ditch. In this case we can write

$$\frac{d}{dt} F_j = -\frac{M f_j}{R_j}, F(t_j) = 1, \tag{3.4}$$

where t_j is the start time of task j. In order to incorporate negotiation into our model, we assume that all agents spend some fraction of time, β, negotiating, where $\beta \in [\beta_0, \beta_1]$. The constants β_0 and β_1 are the minimum and maximum levels of communication overhead, respectively. We assume that agents cannot work on tasks and negotiate simultaneously. Consequently, of each small time increment, Δt, only $(1-\beta)\Delta t$ is available for task completion. The natural modification to (3.4) is therefore

$$\frac{d}{dt}F_j = -(1-\beta)\frac{Mf_j}{R_j}, F(t_j) = 1. \tag{3.5}$$

The negotiation overhead, β, can in principle depend on many factors, including the behavior of the agents. For example assume it is a function only of the number of active tasks N. Let N_0 be the number of tasks that result in half-saturation of the network. The negotiation fraction could be modeled by

$$\beta = \beta_0 + \frac{\beta_1 N^\alpha}{N_0^\alpha + N^\alpha}, \alpha \in \mathbb{N}. \tag{3.6}$$

This function has values of β approaching β_0 for $N \ll N_0$ and values approaching β_1 for $N \gg N_0$, with $\beta = \frac{1}{2}$ at $N = N_0$ and increasing α creating a more abrupt transition from β_0 to β_1. For our analysis we are only concerned with what the resulting level of communication actually is, regardless of how the network operates, and will simply assume $\beta = \beta_0$. To find F_j we need to solve the differential equation (3.5). Since $\beta, M,$and R_j are constants, to finish specifying the model need to know the fraction of resources f_j assigned to task j at any time t.

3.2.3 Resource Allocation Models

Rather than attempt to model some negotiation scheme, we will assume it has a known deterministic outcome. A resource allocation model f_j is a function that will be used in (3.4) to describe what fraction of available resources are allocated to task j as a function of the states of all active tasks and the negotiation process. The models we will present are by no means the only possibilities. In general the tasks are assigned *weights*, where the weights are determined by the context of the problem. The weighting could be determined by prioritizing the tasks, for example. Again considering the radar tracking problem, it would be sensible to assign a higher priority to targets with a high velocity or that threaten more critical targets. Any conceivable weighting strategy would work in our model as long as the weights sum to one.

Democratic Allocation

An obvious solution to the problem of allocating resources is to divide them evenly. In the radar tracking problem discussed in the introduction, if we assume that tracking each of two trains is equally important, simply have half of the available sensors track each target (neglecting other considerations such as sensor range). In their work on the same problem, S. Fitzpatrick and L. Meertens [14] developed a negotiation strategy based on graph coloring that produces an approximately democratic allocation. We mean democratic in the sense of fairness to the participants –

each agent receives an equal share of the resources

$$f_j = \frac{1}{N}, \tag{3.7}$$

where N is the number of currently active tasks. A weighted version of democratic allocation is used to allocate CPU resources in most operating systems.

Crisis Allocation

Another weighting factor could be to give 'critical' tasks more resources. In the screaming generals context a critical task is one that is close to its deadline relative to the other tasks. The resources assigned are distributed according to

$$f_j = \frac{\frac{1}{D_j - t}}{\frac{1}{D_1 - t} + \frac{1}{D_2 - t} + \cdots + \frac{1}{D_N - t}}. \tag{3.8}$$

The fractions $\frac{1}{D_k - t}, k \in \{1, 2, \ldots, N\}$ are a measure of each task's proximity to the deadline, where $\lim_{t \to D_j} f_j = 1$. For instance if $t = 1$ and $D_1 = 3, D_2 = 4, D_3 = 5$ then $f_1 \approx 0.46, f_2 \approx 0.31, f_3 \approx 0.23$, thus giving the highest fraction of resources to the task with the nearest deadline. The idea of giving tasks nearest to deadline the highest priority is used by C.L. Liu and J.W. Layland in their paper on scheduling tasks on a single processor [15].

Opportunistic Allocation

Smaller tasks are relatively easy to finish, and one (opportunistic) approach would be to finish them first. The fractions $\frac{1}{R_k F_k}, k \in \{1, 2, \ldots, N\}$ are a measure of the inverse size of the task, and the smallest tasks will receive the most resources via the following formula:

$$f_j = \frac{\frac{1}{R_j F_j}}{\frac{1}{R_1 F_1} + \frac{1}{R_2 F_2} + \cdots + \frac{1}{R_N F_N}}. \tag{3.9}$$

Again we consider a three-task example with $R_1 = R_2 = R_3 = 1$, where the tasks are all the same size. Let $F_1 = .5$, $F_2 = .4$, and $F_3 = .1$, which results in $f_1 \approx 0.14$, $f_2 \approx 0.17$, $f_3 \approx 0.69$. Here task three is the 'easiest' so it receives the largest proportion of the resources. R. Armstrong [16] describe several opportunistic-type strategies for completing tasks in a distributed computing environment.

3.2.4 Dimensional Analysis and the Critical Start Time

To obtain a dimension-free form of (3.5) we let

$$\tau = (1 - \beta_0)\frac{M}{R_1}t, \delta_j = (1 - \beta_0)\frac{M}{R_1}D_j, \rho_j = \frac{R_1}{R_j}. \tag{3.10}$$

Here τ is the fraction of time elapsed since the start of task one and before its minimum completion time. This implies that it is completed according to

$$\frac{dF_1}{d\tau} = -1, F_1(0) = 1 \Rightarrow F_1(\tau) = 1 - \tau. \tag{3.11}$$

We can see that this task will take one unit of dimensionless time to complete, provided no other task interferes.

A central feature of the screaming generals problem is that the agents have no prior knowledge of the tasks they will be assigned. If they did, the resource allocation problem could presumably be solved using a more sophisticated method. Thus, we will assume that from the agents' point of view the tasks assigned are *random* in start time, deadline, and/or size. In terms of our dimensionless parameters we will say that these parameters have a uniform distribution given by $\tau_j \sim U[0, T]$, $\delta_j \sim U[0, D]$, and $\rho_j \sim U[0, R]$, respectively, where T, D,and R are the maximum values for each parameter. There are several reasons for our choice of a uniform distribution. Uniform distributions are easier to analyze and very easy to simulate. In addition, without any knowledge of the specific tasks the agents will be solving, a uniform distribution is the fairest to use in evaluating the performance of different resource allocation strategies, and gives ample opportunity to test the effect of extreme parameter choices and interactions among extremes. However, in principle there is no reason an arbitrary distribution cannot be used.

Since task parameters are random we need to quantify the probabilities of any events we are interested in. The probability one task will succeed (or fail) in isolation is something we want to know. From a combinatorial perspective there are a huge number of possible events, with various numbers of existing tasks, new tasks, and successful/failed tasks. A possible generalization is to maintain a running count of task probabilities. For instance, suppose we know the probability of one task succeeding in isolation. Then we add a task and find the probability that the addition of a new task will cause the first task to fail. We would then need to find the probability that the new task can be successful given the existence of the first task.

This method can be extended to an arbitrary number of tasks, assuming the deadlines are ordered according to $\delta_1 < \delta_2 < \ldots < \delta_N$, which is reasonable if we allow the tasks to be re-indexed (and since time is a continuous variable $P(\delta_i = \delta_j) = 0$). If this ordering holds then the first task that will fail is task one. So the only new quantities to compute with the addition of task $N + 1$ are the probability that task one will fail and the probability that task $N + 1$ will succeed given the existence of N tasks. This is a simplification – it is possible that one of the tasks will succeed before task one fails, thus reducing the system to N tasks. However, this will increase the available resources, making our previously computed value for the probability of task one failing an *over*-estimate and the probability for task $N + 1$ succeeding an *under*-estimate. This method, then, will consistently under-estimate the probability of successful task completion.

In order to find these probabilities we introduce the concept of a *critical start time*, τ_j^*. Consider the two-task case and assume task one will be successful in isolation, which implies that there exists some time τ such that $\tau \leq \delta_1$and$F(\tau) = 0$, *i.e.* the task will finish before or at the deadline. When task two is introduced at time τ_2 it will cause task one to be completed *later* than it would have in isolation, because task two is now using some of the resources task one was using. It follows that by making τ_2 earlier and using more of task one's resources, we will eventually cause task one to finish exactly at its deadline, meaning that $F_1(\delta_1) = 0$. This value of τ_2 is the critical start time for task two and will be labeled τ_2^*. Now we can formulate the probability of success for task one given the addition of task two as $P(\tau_2 \geq \tau_2^*)$. If task two starts before the critical time it will cause task one to fail by consuming too many of the resources. In general τ_{N+1}^* is the critical start time of the newest task that could cause task one to fail.

Democratic Allocation

If $N = 2$ and the tasks have start times τ_1 and τ_2, we have a linear system

$$\begin{array}{ll} dF_1 = -\frac{M}{2R_1}\, dt = -\frac{1}{2} d\,\tau \Rightarrow & \frac{dF_1}{d\tau} = -\frac{1}{2}, \quad F_1(\tau_2) = 1 - \tau_2, \\ dF_2 = -\frac{M}{2R_2} \cdot \frac{R_1}{R_1}\, dt = -\frac{1}{2}\rho_2\, d\tau \Rightarrow & \frac{dF_2}{d\tau} = -\frac{\rho_2}{2}, \quad F_2(\tau_2) = 1. \end{array} \quad \tau_2 \geq \tau_1 = 0, \tag{3.12}$$

which has the solution

$$F_1 = 1 - \tfrac{1}{2}\tau - \tfrac{1}{2}\tau_2, \quad F_2 = 1 - \tfrac{\rho_2}{2}\tau + \tfrac{\rho_2}{2}\tau_2 \ , 0 \leq \tau \leq 2 - \tau_2. \tag{3.13}$$

The critical start time in this case is

$$F_1(\delta_1) = 0 = 1 - \frac{1}{2}(\delta_1 + \tau_2^*) \Rightarrow \tau_2^* = 2 - \delta_1. \tag{3.14}$$

Consider a specific example where $\rho_2 = 1$, $\delta_1 = \frac{3}{2}, \tau_2 = \tau_2^* = \frac{1}{2}$. This situation is illustrated in figure (3.2).

In general if $N = n$ the system we have the following by induction on (3.12):

$$\begin{array}{ll} \frac{dF_1}{d\tau} = -\frac{1}{n}, & F_1(\tau_n) = 1 - \frac{1}{n-1}\tau_n - \frac{1}{n-1}\tau_{n-1} - \ldots - \frac{1}{2}\tau_2, \\ \frac{dF_2}{d\tau} = -\frac{\rho_2}{n}, & F_2(\tau_n) = 1 + \frac{\rho_2}{n-1}\tau_n - \frac{\rho_2}{n-1}\tau_{n-1} - \ldots - \frac{\rho_2}{2}\tau_2, \\ \frac{dF_3}{d\tau} = -\frac{\rho_3}{n}, & F_3(\tau_n) = 1 + \frac{\rho_3}{n-1}\tau_n - \frac{\rho_3}{n-1}\tau_{n-1} - \ldots - \frac{\rho_3}{6}\tau_3, \\ & \vdots \\ \frac{dF_{n-1}}{d\tau} = -\frac{\rho_{n-1}}{n} & F_{n-1}(\tau_n) = 1 + \frac{\rho_{n-1}}{n-1}\tau_n - \frac{\rho_{n-1}}{n-1}\tau_{n-1}, \\ \frac{dF_n}{d\tau} = -\frac{\rho_n}{n}, & F_n(\tau_n) = 1. \end{array} \quad \delta_1 > \tau_n \geq \ldots \geq \tau_2 \geq \tau_1 = 0, \tag{3.15}$$

All n tasks must start before the first deadline, otherwise the system really has $n-1$ or fewer tasks. Using induction we obtain the solution

$$\begin{array}{l} F_1 = 1 - \frac{1}{n}\tau - \frac{1}{2}\tau_2 - \frac{1}{6}\tau_3 - \ldots - \frac{1}{(n-1)(n-2)}\tau_{n-1} - \frac{1}{n(n-1)}\tau_n, \\ F_2 = 1 - \frac{\rho_2}{n}\tau + \frac{\rho_2}{2}\tau_2 - \frac{\rho_2}{6}\tau_3 - \ldots - \frac{1}{(n-1)(n-2)}\tau_{n-1} - \frac{1}{n(n-1)}\tau_n, \\ F_3 = 1 - \frac{\rho_3}{n}\tau + \frac{\rho_3}{3}\tau_3 - \frac{\rho_3}{12}\tau_4 - \ldots - \frac{1}{(n-1)(n-2)}\tau_{n-1} - \frac{1}{n(n-1)}\tau_n, \\ \vdots \\ F_{n-1} = 1 - \frac{\rho_{n-1}}{n}\tau + \frac{\rho_{n-1}}{n-1}\tau_{n-1} - \frac{\rho_{n-1}}{n(n-1)}\tau_n, \\ F_n = 1 - \frac{\rho_n}{n}\tau + \frac{\rho_n}{n}\tau_n. \end{array} \tag{3.16}$$

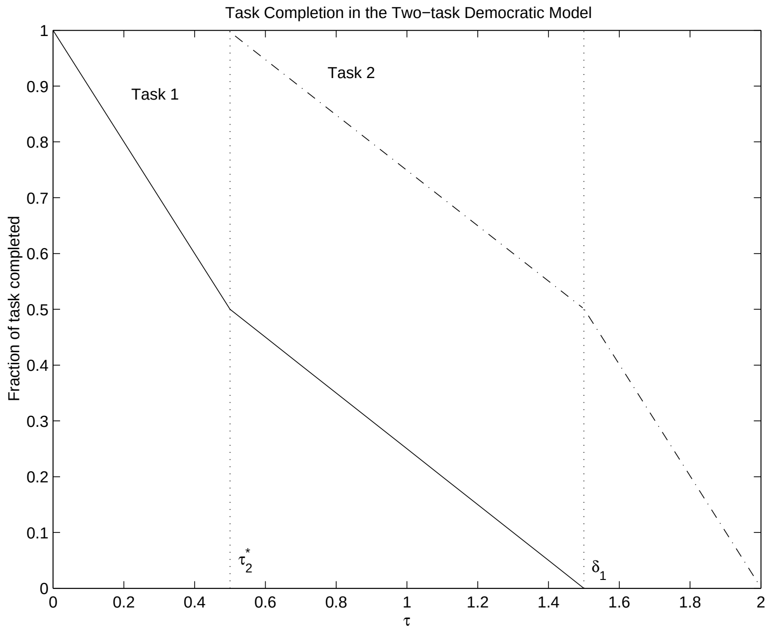


Figure 3.2: Task completion under the democratic resource allocation model. Due to task two starting at its critical time τ_2^*, task one finishes exactly at deadline. The non-dimensional parameters are $\rho_2 = 1$, $\delta_1 = \frac{3}{2}, \tau_2 = \tau_2^* = \frac{1}{2}$.

Setting $F_1(\delta_1) = 0$ gives the critical start time for task n :

$$\tau_n^* = n(n-1) - (n-1)\delta_1 - \frac{n(n-1)}{2}\tau_2 - \frac{n(n-1)}{6}\tau_3 - \ldots - \frac{n}{n-2}\tau_{n-1}. \tag{3.17}$$

This gives us an appreciation for the complicated nature of the solution to these systems, even with an extremely simple resource allocation. At a minimum we must keep track of n initial conditions, each of which changes with the addition or removal of a task. And as we will see the solution can be more difficult or even impossible to obtain in the case of other strategies.

Crisis Allocation

The next most complicated allocation strategy, crisis allocation, generates linear equations, but with non-constant coefficients. We first simplify the crisis equations as follows:

$$\begin{aligned} \tfrac{dF_1}{dt} &= -(1-\beta)\tfrac{M}{R_1}\left(\tfrac{D_2-t}{D_1+D_2-2t}\right), \quad F_1(t_2) = 1-(1-\beta)\tfrac{M}{R_1}t_2, \\ \tfrac{dF_2}{dt} &= -(1-\beta)\tfrac{M}{R_2}\left(\tfrac{D_1-t}{D_1+D_2-2t}\right), \quad F_2(t_2) = 1. \end{aligned} \tag{3.18}$$

Removing t from the numerator to facilitate integration gives

$$\begin{aligned} \tfrac{dF_1}{dt} &= -(1-\beta)\tfrac{M}{2R_1}\left(1+\tfrac{D_2-D_1}{D_1+D_2-2t}\right), \quad F_1(t_2) = 1-(1-\beta)\tfrac{M}{R_1}t_2 \\ \tfrac{dF_2}{dt} &= -(1-\beta)\tfrac{M}{2R_2}\left(1+\tfrac{D_1-D_2}{D_1+D_2-2t}\right), \quad F_2(t_2) = 1 \end{aligned} \quad . \tag{3.19}$$

To non-dimensionalize we multiply by one in several places;

$$\begin{aligned} \frac{dF_1}{dt} &= -(1-\beta)\frac{M}{2R_1}\left(1+\frac{(1-\beta)\frac{M}{R_1}(D_2-D_1)}{(1-\beta)\frac{M}{R_1}(D_1+D_2-2t)}\right), & F_1(t_2) &= 1-(1-\beta)\frac{M}{R_1}t_2, \\ \frac{dF_2}{dt} &= -(1-\beta)\frac{M}{2R_2}\left(\frac{R_1}{R_1}\right)\left(1+\frac{(1-\beta)\frac{M}{R_1}(D_1-D_2)}{(1-\beta)\frac{M}{R_1}(D_1+D_2-2t)}\right), & F_2(t_2) &= 1. \end{aligned} \tag{3.20}$$

which results in the dimensionless form

$$\begin{aligned} \frac{dF_1}{d\tau} &= -\frac{1}{2}\left(1+\frac{\delta_2-\delta_1}{\delta_1+\delta_2-2\tau}\right), & F_1(\tau_2) &= 1-\tau_2 \\ \frac{dF_2}{d\tau} &= -\frac{\rho_2}{2}\left(1+\frac{\delta_1-\delta_2}{\delta_1+\delta_2-2\tau}\right), & F_2(\tau_2) &= 1. \end{aligned} \quad \tau_2 \geq \tau_1 = 0, \tag{3.21}$$

Solving for F_1 and F_2 gives

$$\begin{aligned} F_1 &= 1-\frac{1}{2}\left[\tau+\tau_2+\frac{\delta_2-\delta_1}{2}\log\left(\frac{\delta_1+\delta_2-2\tau}{\delta_1+\delta_2-2\tau_2}\right)\right], \\ F_2 &= 1-\frac{\rho_2}{2}\left[\tau+\tau_2+\frac{\delta_1-\delta_2}{2}\log\left(\frac{\delta_1+\delta_2-2\tau}{\delta_1+\delta_2-2\tau_2}\right)\right]. \end{aligned} \tag{3.22}$$

We observe that $F_1(\delta_1) = 0$ is transcendental in τ_2, making it impossible to find τ_2^* in closed form. A one-term Taylor approximation of the log term allows us to find

$$\tau_2^* = \frac{1}{1+\delta_1+\delta_2}\left[2-\delta_1+\frac{\delta_2-\delta_1}{2}\log\left(\frac{\delta_2-\delta_1}{\delta_2+\delta_1}\right)\right], \tag{3.23}$$

which is valid when $2\tau_2 \ll \delta_1+\delta_2$.

While it is possible to solve the n-task equation, it becomes increasingly difficult to simplify the fractions as n increases. And as we saw with the democratic allocation, the initial conditions are complicated as well. Given the increasing degree of analytic complexity, it is doubtful that direct characterization of τ_j^* and calculation of $P(\tau_j \geq \tau_j^*)$ to assess success probabilities will be more illuminating than direct simulation.

Opportunistic Allocation

Potentially the most complicated allocation strategy is opportunistic, with model equations which are fully non-linear. Simplifying the opportunistic equations (3.5) and (3.9) with $n = 2$ gives

$$\begin{aligned} \frac{dF_1}{dt} &= -(1-\beta)\frac{M}{R_1}\left(\frac{R_2F_2}{R_1F_1+R_2F_2}\right), & F_1(t_2) &= 1-(1-\beta)\frac{M}{R_1}t_2, \\ \frac{dF_2}{dt} &= -(1-\beta)\frac{M}{R_2}\left(\frac{R_1F_1}{R_1F_1+R_2F_2}\right), & F_2(t_2) &= 1. \end{aligned} \tag{3.24}$$

Factoring yields

$$\begin{aligned} \frac{dF_1}{dt} &= -(1-\beta)\frac{M}{R_1}\frac{R_2}{R_2}\left(\frac{F_2}{\frac{R_1}{R_2}F_1+F_2}\right), & F_1(t_2) &= 1-(1-\beta)\frac{M}{R_1}t_2, \\ \frac{dF_2}{dt} &= -(1-\beta)\frac{M}{R_2}\frac{R_1}{R_1}\left(\frac{F_1}{F_1+\frac{R_2}{R_1}F_2}\right), & F_2(t_2) &= 1. \end{aligned} \tag{3.25}$$

Applying our parameter definitions to the opportunistic allocation gives

$$\begin{array}{ll} \frac{dF_1}{d\tau} = -\frac{F_2}{\rho_2 F_1 + F_2}, & F_1(\tau_2) = 1 - \tau_2, \\ \frac{dF_2}{d\tau} = -\frac{\rho_2 F_1}{F_1 + \frac{1}{\rho_2} F_2} = -\frac{\rho_2^2 F_1}{\rho_2 F_1 + F_2}, & F_2(\tau_2) = 1. \end{array} \tag{3.26}$$

Using *Maple* we find the solution to be

$$\left. \begin{array}{l} F_1 = -\frac{1}{2}\left(\frac{\rho_2\tau^2 - 2(1+\rho_2)\tau + 2\rho_2 - 2\tau_2\rho_2 + \tau_2^2\rho_2 + 2}{\rho_2\tau - 1 - \rho_2}\right), \\ F_2 = -\frac{1}{2}\left(\frac{\rho_2^2\tau^2 - 2\rho_2(1+\rho_2)\tau + 2\rho_2 + 2\tau_2\rho_2^2 - \tau_2^2\rho_2^2}{\rho_2\tau - 1 - \rho_2}\right). \end{array} \right. \delta_1 \geq \tau \geq \tau_2, \tag{3.27}$$

Setting $F_1(\delta_1) = 0$ we can solve for τ_2^* using the quadratic formula to obtain

$$\tau_2^* = \rho_2 \pm \sqrt{\rho_2^2 - 2\rho_2 - 2\delta_1\rho_2 - \delta_1\rho_2^2 + 2\delta_1 + 2}. \tag{3.28}$$

We will use the earlier of the two times, unless one is negative or imaginary, then we will use the non-negative one. As with the crisis model, the solution of the n-task system of equations is increasingly difficult to obtain and decreasingly informative.

In this section we have shown a method for obtaining the critical start time for the last task, which is the minimum information necessary to compute the probability a task will succeed or fail, in the 'worst' case. We will now use these τ_j^* to find bounding probabilities for task failure.

3.3 Task Completion Probabilities

3.3.1 One Active Task

We have a condition for task failure determined by (3.11): if the deadline δ_1 is less than one, the task will fail. Otherwise, it will succeed. Suppose δ_1 is random according to some probability density function p_δ such that $\delta_j \in [0, D]$ where $D > 1$ (to guarantee a non-zero probability). Then the probability that the task is a success is

$$P(\delta_1 > 1) = \int_1^D p_\delta(s)\text{ds}. \tag{3.29}$$

For example, if we let p_δ be uniform on $[0, 2]$ then (3.29) evaluates to

$$\int_1^2 \frac{1}{2}\text{ds} = \frac{1}{2}, \tag{3.30}$$

which is an intuitive result since it is equally likely for the deadline to come before or after $\tau = 1$.

3.3.2 Two Active Tasks – Democratic

Let us assume that the first task will succeed if left to its own devices, *i.e.,* $\delta_1 \geq 1$. With two tasks running under the democratic regime we compare the start time of the second task τ_2 to the critical start time τ_2^* we obtained in (3.14). If $\tau_2 < \tau_2^*$ then task one will fail due to too many resources being consumed by the other task. Suppose τ_2, the actual, start time, is randomly distributed with density p_τ with $\tau \in [0, T]$ where $2 \geq T > 0$. Then the probability of success is

$$P(\tau_2 > \tau_2^*) = P(\tau_2 > 2 - \delta_1) = \int_{2-\delta_1}^{T} p_\tau(s)\text{ds}. \tag{3.31}$$

Again considering a uniform probability on $[0, 2]$ for p_τ gives a probability of success

$$\int_{2-\delta_1}^{2} \frac{1}{2}\text{ds} = \frac{\delta_1}{2}. \tag{3.32}$$

This result is intuitive if we keep in mind it is conditioned on the success of the first task in isolation. If $\delta_1 = 1$ then the probability of the first task succeeding is now $\frac{1}{2}$. If $\delta_1 = 2$ then the probability of the first task succeeding is one, since it will take two units of dimensionless time to complete in the worst case ($\tau_2 = 0$).

3.3.3 Two Active Tasks – Crisis and Opportunistic

Under the crisis model we found that it is not possible to solve for τ_2^* explicitly. However, given a value for δ_1 we can find the level curve associated with $F_1(\delta_1) = 0$. We can then integrate over the region where $\tau_2 > \tau_2^*$. For example of a level curve with $\delta_1 = 1$, $\delta_2 \sim U[1, 7]$ and $\tau_2 \sim U[0, 2]$ results in a probability of success of approximately 0.3175 using Monte-Carlo integration. Applying the same parameters and method to the opportunistic case with $\rho_2 \sim U[0, 1]$ we find a success probability of 0.1826.

For the two-task case we needed to use numerical methods – one to find the level curve and one to find the area of the region. With three or more active tasks we would need to find a level surface and then integrate over the proscribed volume, in three-dimensional or higher space. As we continue to add tasks, computing these probabilities becomes more and more expensive in terms of processor time. In addition, our analytical methods become much more difficult with additional tasks. The combination of these two factors leads us to attempt a simulation of a series of random tasks and analyze the numerical results, as opposed to the more expensive (and cumbersome) numerical realization of analytic results.

3.3.4 Numerical Simulation

In our numerical simulation we return to the dimensional model in equation (3.5). For each resource allocation strategy we will simulate random populations of tasks and measure the number of failures and successes in each. After running a large number of these simulations the average proportion of successful tasks for each strategy will be determined, providing us a metric for comparing the efficiency of the strategies. We will also vary deadline, communication overhead, and task loading in order to make some conclusions concerning the dynamics of the screaming generals problem. The values of the other parameters will be arbitrary. The average proportion of successful tasks is plotted for all three strategies over a wide range of task densities. A sample run is shown in figure 3.3.

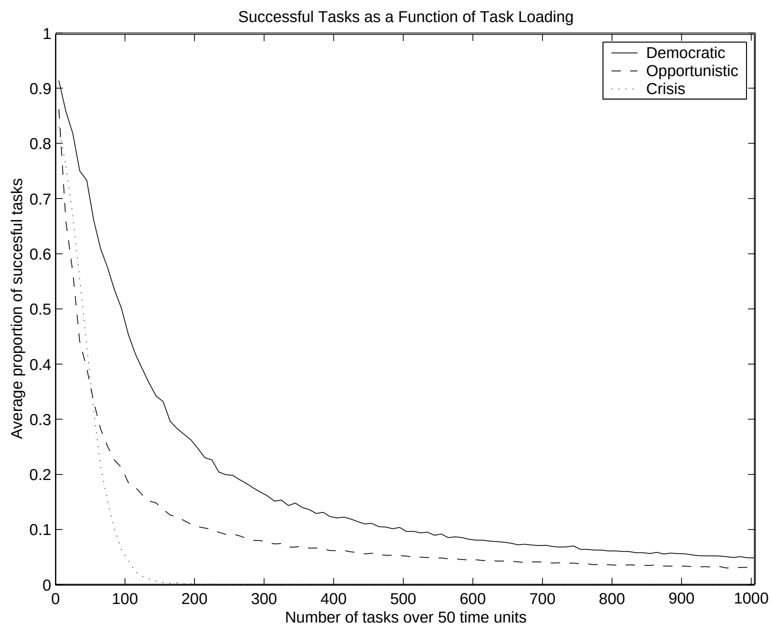


Figure 3.3: Average proportion of successful tasks over 60 simulations per task loading (horizontal axis) plotted by resource allocation strategy. The solid, dashed, and dotted lines are the democratic, crisis, and opportunistic values, respectively. Note the point where the crisis curve crosses the opportunistic curve.

The predicted values from the previous sections are plotted in approximately the correct location. If there are two tasks in a two time unit region, it follows that, on average, the task loading is approximately 50 over a 50 time unit duration. For each task loading, N, over 50 time units, a random vector of N start times is chosen uniformly on $[0, 50]$ with associated deadlines chosen uniformly on $[\text{Start}, \text{Start}+D]$ where D is some maximum deadline. Any deadline which is greater than 50 is set to 50. The system of ODE's described in (3.5) is solved using the 4^{th} order Runge-Kutta method. The status of each task at its deadline is assessed, and the total number of successes and failures is recorded.

To characterize simulation output we need a fit which varies smoothly from zero to one as the input varies from one to infinity. In addition, initial explorations indicate power-law behavior in the tails of the results. We therefore choose

$$\pi = \frac{kx^b}{1 + kx^b} \tag{3.33}$$

This page intentionally left blank

as the model function, where π is the proportion of successful tasks, x is the task loading, and k and b are constants. This model is equivalent to

$$\log\left(\frac{\pi}{1-\pi}\right) = a + b\log x, \tag{3.34}$$

where $k = e^a$. Using linear regression on this model we can determine k and b in (3.33) for each average success curve. The model fit to the curves in figure 3.3 is shown in figure 3.5. The correlation coefficient values are $r^2 \approx 0.99$ for democratic, $r^2 \approx$0.99 for opportunistic, and $r^2 \approx$0.94 for crisis.

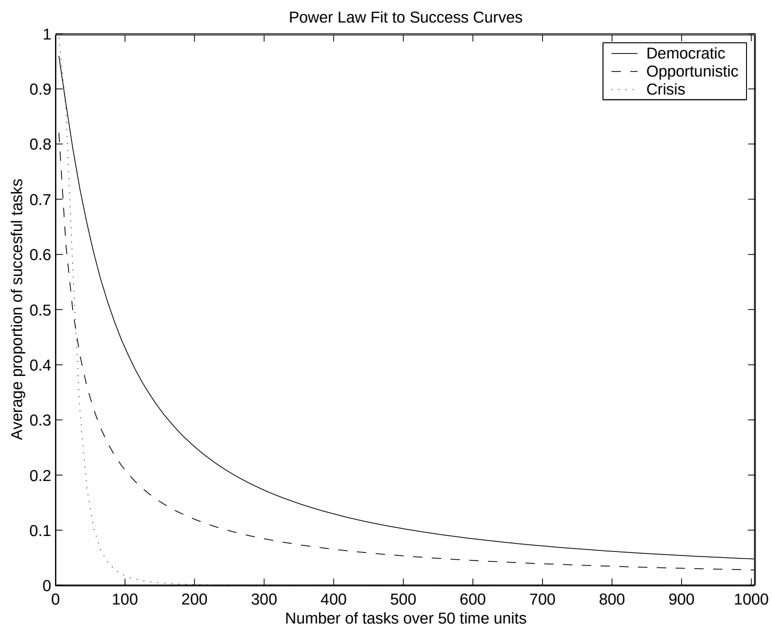


Figure 3.5: Power law fit to the success curves in figure 3.3, using the equation described in the text. The communication overhead is $\beta = 0$, the maximum deadline is $D = 6$, and the tasks randomly have sizes between 0 and 100 percent of the available resources. This fit was designed to match the true values asymptotically with increasing task size. Consequently it matches the true values well over most of the domain, but in the 0 to 50 region it overestimates the success probabilities.

Our model also diagnoses the asymptotic rate of decay of the average success proportion with increasing task load to be $\mathcal{O}(x^b)$. In figures 3.6 and 3.7 we allowed the communication time β to vary between 0 and 0.95, and the maximum deadline to vary between 1 and 20, respectively when conducting the simulations. The model fit exponent b is plotted against these parameters.

From these plots we can see that both the democratic and opportunistic strategies are fairly consistent and behave approximately as $\mathcal{O}(x^{-1})$. The crisis strategy is more erratic and decays approximately in the $\mathcal{O}(x^{-2.5})$ to $\mathcal{O}(x^{-3})$ range. Clearly crisis is the asymptotic loser in this

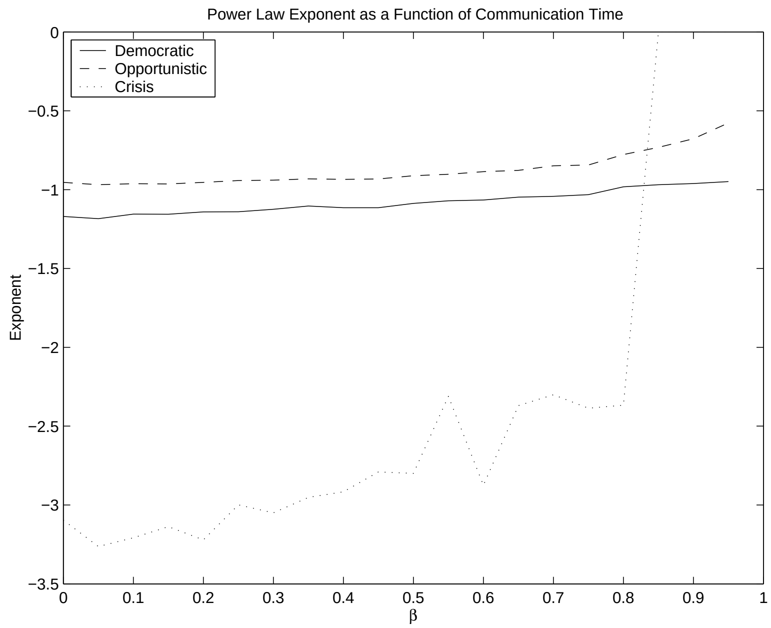


Figure 3.6: Plot of power law exponents against communication overhead. The democratic and opportunistic strategies decay consistently over different communication overheads approximately on the order of x^{-1}. Crisis decays erratically over the domain between the orders of x^{-3} and x^{-2}. The sharp increase at $\beta = 0.85$ occurs not because the strategy gets better, but because it no longer completes any tasks, and therefore generates a zero fit parameter. Crisis has the worst asymptotic behavior of the three strategies.

analysis, although there is a range of low task loading in which it performs slightly better than the opportunistic model, as shown in figure 3.3. Also, it would appear from figures 3.6 and 3.7 that opportunistic allocation is asymptotically better than democratic, since it has the exponent smallest in magnitude. But in figure 3.3 we see that it is uniformly worse than democratic in this case. This is not necessarily a contradiction as the decay exponent does not take into account the vertical-axis intercepts of the curves. As we will see there are regions of the parameter space (usually at high task loads) in which opportunistic does out-perform democratic by a small margin, as well as regions (very low task loads) in which crisis does better than opportunistic.

Generally, for any choice of simulation parameters, as load increases there is first a regime in which democratic is superior, followed by opportunistic. Crisis starts out better than opportunistic, then switches places at a fairly low task density. Characterizing these switches is the object of the next section.

3.3.5 Crossover Points

Figures 3.3 and 3.5 illustrate what we will call a ‘crossover point,’ where two average success curves cross. We see that the crisis strategy is superior to the opportunistic strategy until it reaches

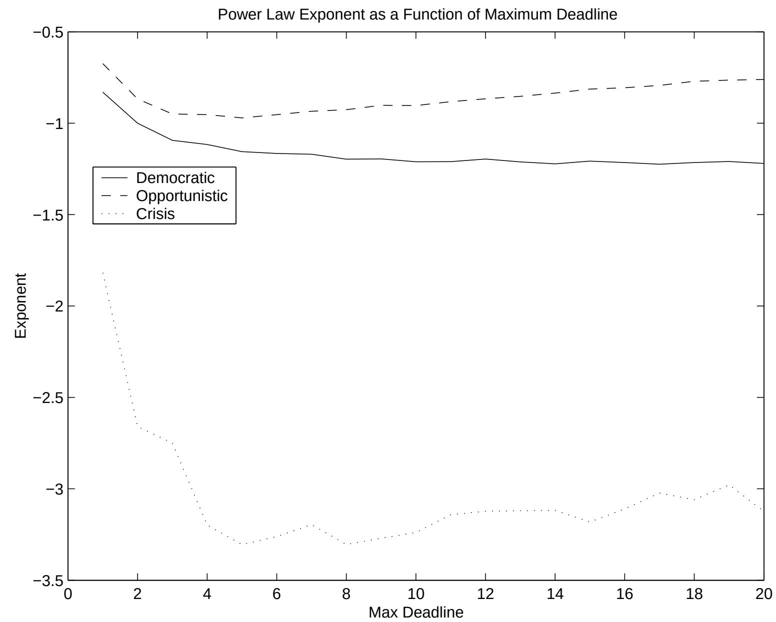


Figure 3.7: Plot of power law exponents against maximum deadline. This is essentially the inverse of figure 6. This is because tasks are easier with increasing deadline, as opposed to harder with increasing communication time. Again, we see that the crisis strategy has the worst asymptotic behavior over the domain.

a crossover point after which the opportunistic strategy is better. From the same data used to obtain the power law exponent in figures 3.6 and 3.7 we measured this crossover point to the nearest five-task unit on both the actual curve and our fitted curve. These points are plotted in figures 3.8 and 3.9.

As β increases or the maximum deadline D decreases, opportunistic eventually outperforms democratic. This is shown in figures 3.10 and 3.11. In these figures the predicted values match the actual values much closer than in figures 3.6 and 3.7. This is due to the model we used – we wanted it to match the actual results in the high task-loading regime, in order to find the decay exponents. Consequently it does not match as well in the low task-loading regime where the crisis-opportunistic crossover occurs. For any two predicted curves with fit parameters k_1 and k_2, and exponents b_1 and b_2, the crossover location in terms of task loading is given by

$$x = \left(\frac{k_2}{k_1}\right)^{\frac{1}{b_1 - b_2}}. \tag{3.35}$$

We conclude that there are regions in the parameter space in which the democratic method is the most successful, and other regions where opportunistic is better. The regions where opportunistic is superior have difficult tasks due to significant communication overhead or short deadlines, and heavy task loads. The crisis strategy is nowhere superior to the democratic, and this fact in addition to its larger decay exponent make it definitively the worst of the three strategies. This should not

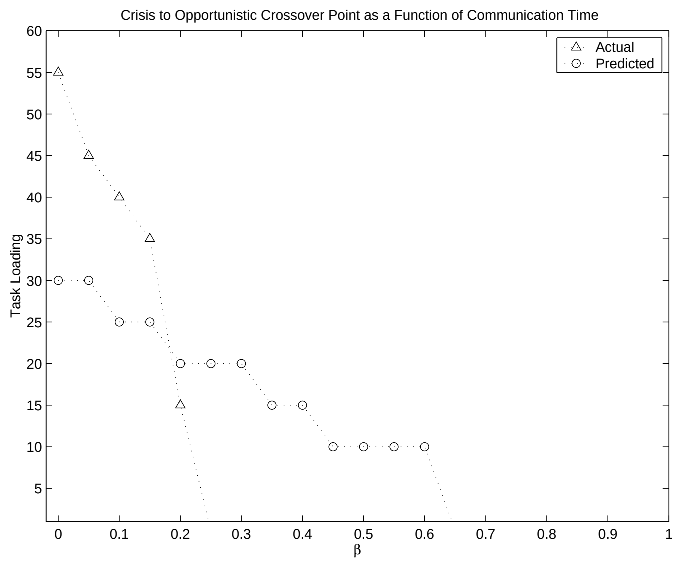


Figure 3.8: Crossover points where the opportunistic strategy becomes superior to the crisis strategy as a function of communication overhead β, where the max deadline is $D = 6$. Both the actual crossover points and the points predicted by our power law fit are plotted. As mention in the text, the fitted model is designed to accurately predict the asymptotic behavior of the strategies with increasing task density. These crossover points occur at small task loads, where the model does not fit as accurately. Consequently, the predicted values are not very accurate.

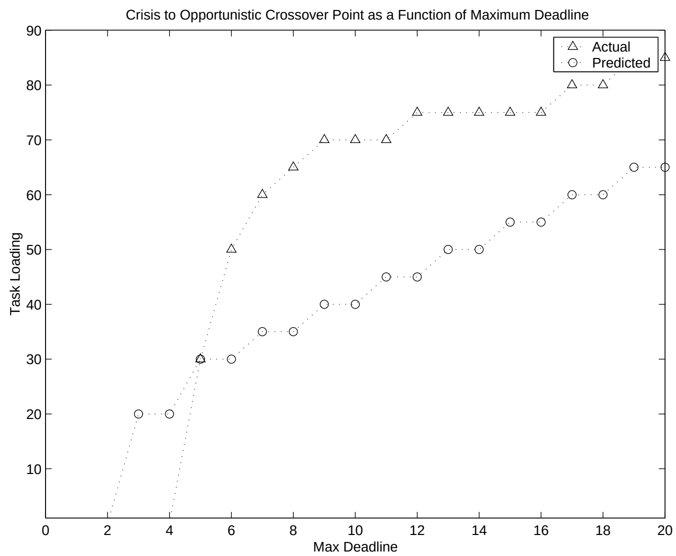


Figure 3.9: Crossover points where the opportunistic strategy becomes superior to the crisis strategy as a function of maximum deadline D, where the communication overhead is $\beta = 0$. Both the actual crossover points and the points predicted by or power law fit are plotted. As mention in the text, the fitted model is designed to accurately predict the asymptotic behavior of the strategies with increasing task density. These crossover points occur at small task loads, where the model does not fit as accurately. Consequently, the predicted values are not very accurate.

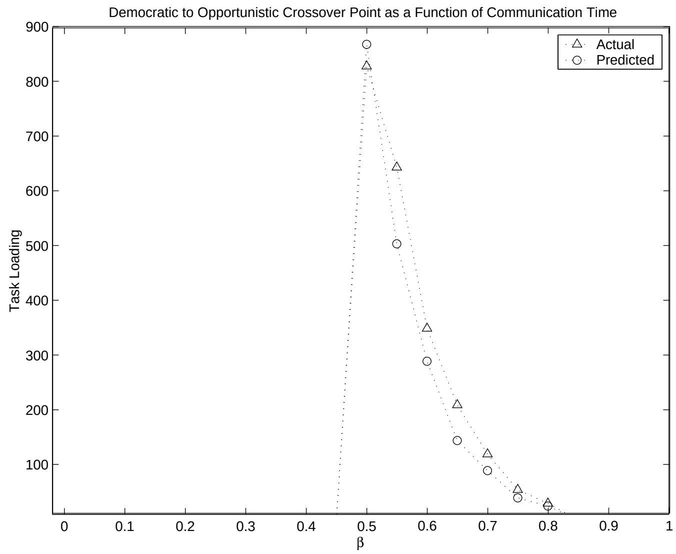


Figure 3.10: Crossover points where the opportunistic strategy becomes superior to the democratic strategy as a function of communication time β, where the max deadline is $D = 6$. Both the actual crossover points and the points predicted by or power law fit are plotted. As mention in the text, the fitted model is designed to accurately predict the asymptotic behavior of the strategies with increasing task density. These crossover points occur at large task loads, where the model does not fit as accurately. Consequently, the predicted values are accurate. The sharp increase is due to the fact that no crossover occurs before $\beta = 0.5$.

be too surprising as common sense dictates against leaving things until the last minute.

Furthermore, increasing β seems to generate results that are the inverse of increasing D. If we look at the dimensionless deadline $\delta_j = (1-\beta)\frac{M}{R_1}D_j$, we can see the reason for this. Increasing β makes δ smaller, thus increasing the probability of task failure, whereas increasing the maximum deadline D will increase δ on average. Using the same reasoning it seems obvious that increasing the number of resources M will make things easier, while increasing the maximum task size R will make things harder. This also squares with our intuition about the problem.

3.4 Conclusion

In this chapter we have demonstrated a framework for analyzing a large-scale system with multiple independent agents. Our analysis is independent of the type of task being performed, as long as that task is homogeneous with respect to resource use (each member of our pool of M men is equally skilled) and difficulty (each section of ditch is equally difficult). The actual procedure for negotiation can also be ignored if it can be cast in the form of a resource allocation strategy with an associated characterization of the communication and negotiation costs (β). While our analytical methods are eventually bogged down by increasing the number of tasks, they do provide valuable

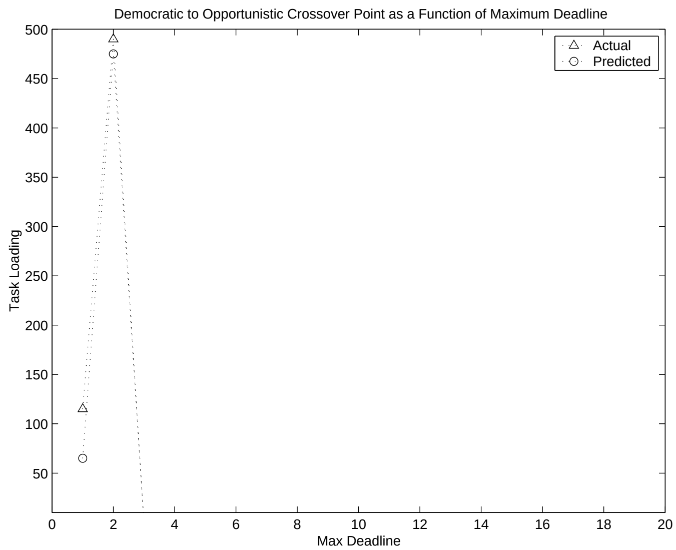


Figure 3.11: Crossover points where the opportunistic strategy becomes superior to the democratic strategy as a function of maximum deadline D, where the communication overhead is $\beta = 0$. Both the actual crossover points and the points predicted by or power law fit are plotted. As mention in the text, the fitted model is designed to accurately predict the asymptotic behavior of the strategies with increasing task density. These crossover points occur at large task loads, where the model does not fit as accurately. Consequently, the predicted values are accurate. The sharp decrease is due to the fact that no crossover occurs after $D = 3$.

insight into the structure and complexity of the problem.

We might also consider applying our results to task completion in real life. In general, democratic allocations Thperform well – in the real world one could apply this by giving equal time to all tasks, all other things being equal. From the authors' experience, this is perhaps an unrealistic, 'ideal' strategy. Usually pressure from deadlines forces one to adopt either a crisis or an opportunistic allocation. Our results show that crisis can be a successful strategy at low task densities. But as the number of tasks increases, someone operating under a crisis management strategy becomes increasingly stressed and inefficient. Opportunistic, on the other hand, is nearly as effective as democratic, and under very large task densities becomes more efficient. If someone has more work than can possibly be completed on time, it makes sense to finish the smallest tasks. While we would be unjustified to make broad conclusions about the real world from our results, the results do seem to square with our intuition about how people get things done under time constraints.

The results of our simulations show that on average, either the democratic or the opportunistic strategy outperform the other two strategies depending on the parameters chosen. Generally an opportunistic allocation becomes beneficial at higher task loads, when the only tasks that have a good chance to finish are the smaller ones. Another research goal might be to discover the optimal strategy for a given set of parameters. The existence and movement of crossover points also suggests that there are some interesting dynamics in this problem that could be further explored.

Another option is to explore an extension of this problem to the case where each ditch is located at a specific point in space, and there are costs associated with moving resources from one ditch to another. Dependencies among tasks are an additional complication – in the real world often tasks must be completed in a particular order. Logistics problems might be a good test bed for this analysis. Our methods could also be applied to the challenge problem discussed in the introduction, where the resources have a fixed location and the tasks, or targets, move through the system.

3.5 A Track Quality Model for Distributed Sensing Networks

3.5.1 Introduction

A network of distributed sensing and/or computing resources provides an obvious advantage for military applications – it can continue to function if some or possibly even most of the resources are destroyed. In this section we present a model for estimating the ability of the sensors to track a target as a function of the amount of time the sensors spend negotiating. Necessarily we make some assumptions about the behavior of the sensors that do not correspond to any real-world hardware. We introduce a measure of quality, Q, that is relative to a perfectly accurate track. This quality measure has no physical meaning – rather, it allows us to make qualitative judgments about the amount of time the sensors should spend negotiating.

3.5.2 Parameters and Variables

Quantity	Units	Description
j	–	Target index
R	length	Detection radius of a sensor (assumed to be the same for all sensors)
$\vec{x_j}$	–	Position of target j in two-dimensional Euclidian space
A_j	length^2	Area in which any sensor can detect target j
$\mathcal{I}_j$	–	$\{k \| A_j$ intersects $A_k\}$ Note: The $\mathcal{I}_j$ are not unique, *e.g.* if A_1 intersects A_2, $\mathcal{I}_1 = \mathcal{I}_2 = \{1, 2\}$
A_j^I	length^2	Area($\mathcal{I}_j$), the area in which every target in $\mathcal{I}_j$ can be detected
$S(a)$	sensors/length^2	Sensor density over some area a
M_j	sensors	$S(A_j^I) * A_j^I$, number of sensors that can track j and another target
ω	–	Amount of time spent in negotiation and associated overhead per Δt
$f_j(\omega)$	–	Fraction of M_j allocated to j as a function of negotiation time
m_j	sensors	$S(A_j)(A_j + (f_j - 1)A_j^I)$, number of sensors allocated to j after negotiation
$Q_j(t)$	quality	Quality of track j as a function of time
$q_j(m_j)$	quality/time	Quality added to track j per Δt as a function of the number of sensors allocated
λ	1/time	Quality degradation factor

3.5.3 Modeling and Assumptions

The following rate equation is used to model the change in track quality over time:

$$\frac{dQ_j}{dt} = q_j(m_j)(1 - \omega) - \lambda Q_j(t). \tag{3.36}$$

So the track quality is altered at any time by the quality added and the quality lost, being the first and second terms of (3.36) respectively. The quality added is controlled directly by the time spent negotiating ω, and indirectly by the results of negotiation m_j. Quality lost is the current quality multiplied by the degradation factor λ.

We will assume the sensor density is fairly dense and uniform, so the density is a constant $S \approx S(a) \forall a$. This assumption may or may not be reasonable for actual hardware. The area of intersection A_j^I is a function of the $\vec{x_j}$'s, which are assumed to be known with perfect accuracy at any time. In the real world there would likely be some error, but this would be an engineering problem specific to the hardware in question.

So the only variable within control of the system is ω. Our idea is that f_j increases as ω increases, since more time spent negotiating should result in more sensors allocated to the target. However, no specific behavior of this function is required by the model, so any function could be used. As an empirical example, consider the results of a graph coloring algorithm outlined in a technical report by the Kestrel Institute (May 2001). This algorithm implements a democratic allocation – each target receives an equal share of the system resources. However, it takes time to distribute the sensors correctly. Their results follow a curve that looks approximately like

$$f_j(\omega) = \frac{\omega}{n(c+\omega)} \tag{3.37}$$

where n is the number of intersecting targets and c is a constant.

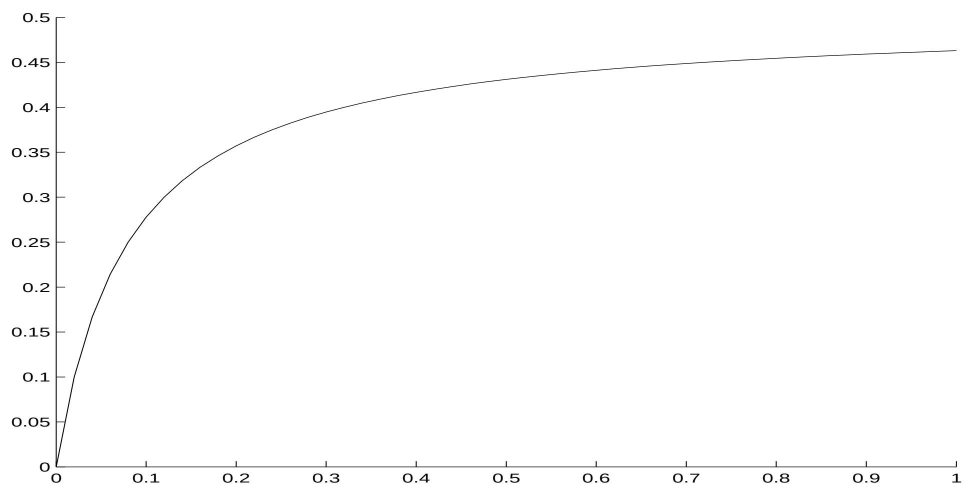


Figure 3.12: $f_j(\omega), n = 2, c = 0.08$

Another hardware specific question is the incremental improvement in quality per sensor allocated to the target. Again, any model can be used. One possibility is that a certain critical quantity of sensors, say μ, is required to get a decent track, and any more than that results in progressively smaller increases. This could be modeled by

$$q_j(m_j) = \frac{m_j^2}{\mu^2 + m_j^2}. \tag{3.38}$$

In the following example we will use a linear model, $q_j(m_j) = m_j$, to exaggerate the benefits of negotiation.

Also there are a variety of functions one could use for the degradation factor λ. The simplest function would be a constant $\lambda = \frac{1}{t_c}$ where a measurement must be taken every t_c seconds or quality will be lost. Other reasonable functions could depend on target variables such as velocity and size.

3.5.4 A Two Target Example

The distance between the two targets is given by $D = \|\vec{x_1} - \vec{x_2}\|_2$. Since there are only two targets we can drop the subscripts and refer to the area of intersection as A^I. It is easily shown that

$$A^I(D) = \begin{cases} 2R^2 cos^{-1}(\frac{D}{2R}) - \frac{D^2}{2}\sqrt{\frac{4R^2}{D^2} - 1}, & D < 2R, \\ 0, & D \geq 2R. \end{cases} \tag{3.39}$$

The targets will follow trajectories $\vec{x_1} = (t, 1)$ and $\vec{x_2} = (t, 1 + 2R(1 - \frac{t}{T}))$ for $0 \leq t \leq T$, using a sensing radius $R = 1$.

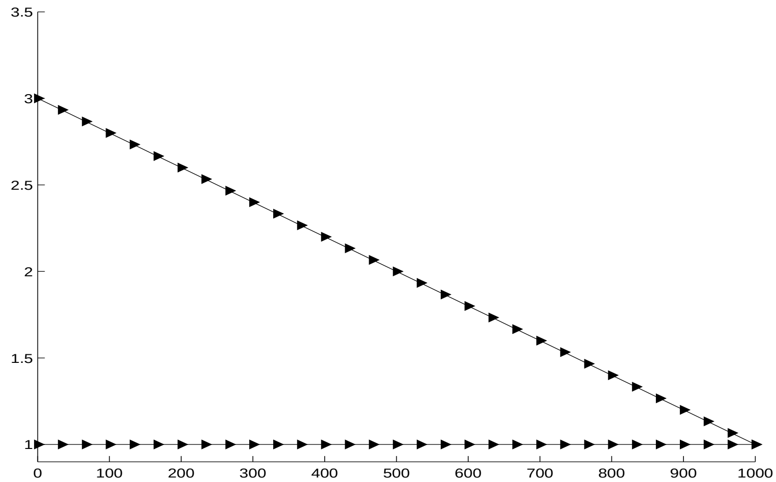


Figure 3.13: $\vec{x_1}, \vec{x_2}, T = 1000$

We define the normalized area of intersection to be $A_N \doteq \frac{A^I}{\pi R^2}$. So over time T, $0 \leq A_N \leq 1$. If we allow T to be very large, A_N will increase slowly from 0 to 1

Note that in the two target case, equation (3.36) is the same for each target, so we will be dropping the subscripts entirely . Then $m(t, \omega) = S(\pi R^2 - (1 - f(\omega))A^I(t))$ so $q(m)$ is a function of t and ω. After the above assumptions and definitions are incorporated we have

$$\dot{Q} = q(t, \omega) - \lambda Q(t). \tag{3.40}$$

Solving for Q yields

$$Q(t, \omega) = e^{-\lambda t} \int_0^t e^{\lambda s} q(s, \omega)(1 - \omega) ds. \tag{3.41}$$

For sufficiently large T, (3.41) should reach equilibrium values over the range of values for A_N. In order to prove this we would have to introduce a slowly varying parameter ϵt and find the solution to (3.41) as $\epsilon \to 0$. An optimal value for ω can then be determined in the limit as $t \to \infty$.

This page intentionally left blank

Chapter 4

Praxeic Decision Theory: Single and Multiple Agents, and Examples

4.1 Overview to Praxeic Decision Theory

Praxeic utility theory is an approach to decision making and control that provides locality of decisions and avoids over proscription by providing set-valued solutions. As opposed to conventional optimization approaches to decision making and control (approaches leading to Bayes decisions, optimal control, optimal filtering, and a host of other successful techniques), the praxeic viewpoint is to weigh each alternative in the decision space on the basis of its own merits, retaining as candidate choices all those whose utility toward approaching a decision goal exceeds the weighted cost of the choice. By considering choices on the basis of individual merit, an optimal choice is not deliberately sought, but candidate choices can be regarded as being good enough for the solution. Thus, praxeic utility theory provides a generally constructive way of approaching problems while breaking out of the "grip of optimality."

The basic framework for praxeic utility theory builds upon two functions satisfying the axioms of probability, which respectively measure the utility of a decision with respect to moving toward some desired goal, and the cost associated with that decision. These functions are called the selectability and the rejectability, and are usually denoted $p_S(u)$ and $p_R(u)$, where the argument u represents a choice under consideration. Out of a set of possible decisions U, the praxeic decision theory indicates that retaining all those choices u for which $p_S(u) \geq bp_R(u)$ are satisficing. In this expression, the parameter b represents a decision maker's "boldness" in rejecting options in the interest of being more selective. Lowering the boldness results in retaining more decisions (being less decisive).

While a straightforward concept, praxeic decision theory has proven successful at a variety of problems, some of which have eluded prior solution. As an example, the inverted pendulum problem — the problem of balancing a broomstick on your palm — has been solved, even for the case when the pendulum (broom) is hanging down. In this case, the control problem is a nonlinear problem not amenable to any standard method.

Notwithstanding its potential for single-agent decision making, one of the real attractions of praxeic utility theory is that it can be rationally extended to group decision making problems. In this realm, it provides a useful new perspective to contrast with prior techniques where each agent acts essentially in a substantively rational way, maximizing its own utility, effectively shutting out solutions which might be useful when viewed from the group perspective. The extension to group decision making arises by defining a joint selectability-rejectability function $p_{\mathbf{S},\mathbf{R}}(u_1, u_2, \ldots, u_N, v_1, v_2, \ldots, v_N)$, with argument slots for the selectability and rejectability of each of the N agents coordinating in the system.

The joint selectability-rejectability function incorporates knowledge about how others' choices may affect the selectability and rejectability of an individual. Since by design the selectability and rejectability functions have the properties of probabilities, factorization of the joint function into units which identify localized behaviors is possible. Thus, it is possible to encode, by means of conditional probabilities, if-then type statements: If an agent Y does "this," then the response of agent X should be "that."

By representing explicitly the influence that other agents may have on their own decisions, the joint selectability rejectability function provides immediately the means for coordinated decision making. When the group wants to act collectively, a joint selectability and a joint rejectability are computed (using the properties of probabilities) by

$$p_{\mathbf{S}}(\mathbf{u}) = \sum_{\mathbf{v}} p_{\mathbf{SR}}(\mathbf{u}, \mathbf{v})$$

and

$$p_{\mathbf{R}}(\mathbf{v}) = \sum_{\mathbf{u}} p_{\mathbf{SR}}(\mathbf{u}, \mathbf{v})$$

where the sum is over the cross section of choices. Using these joint functions, the set of group decisions is selected for which

$$p_{\mathbf{S}}(\mathbf{u}) \geq b p_{\mathbf{R}}(\mathbf{u}).$$

Another approach is for agents to make individual decisions, which still recognize and account for the preferences of the other agents. To do this, an agent can compute its own marginal selectability and rejectability functions by

$$p_{S_i}(u_i) = \sum_{u} p_{\mathbf{S}}(u_i, u)$$

$$p_{R_i}(u_i) = \sum_{u} p_{\mathbf{R}}(u_i, u),$$

where the sums are over all other agents' decision spaces. Then the single-agent decision, accounting for others, is to retain all choices u_i for which

$$p_{S_i}(u_i) \geq b p_{R_i}(u_i)$$

This distinction between a group decision — based on joint selectability and rejectability — and an individual decision — based on marginal selectability and rejectability — leads to a definition for negotiation. It may be recognized that under the constraint of approaching a generally accepted collective decision, negotiators are usually more concerned with meeting minimum performance requirements than with meeting maximum performance, that negotiations should lead to decisions that are both good enough for the group as a whole and also good enough for each individual, and that an element of search is typically necessary as negotiators work toward solutions (Stirling, p. 163). In light of these observations, discrepancies initially evident between individuals based on their marginal selectability and rejectability and the group, based on joint selectability and rejectability, which represent a mismatch between individual and collective choice, can be worked out by a general lowering of expectations. This can be accomplished algorithmically by lowering of boldness, until individual and group preferences share a nonempty set of solutions.

Multi-agent praxeic utility also offers other alternatives to group decision making. For example, it is possible to model a deliberative process: agents making choices based on the choices they think other agents might be making. Explicit knowledge models can be incorporated, with one agent modeling the expected behavior of other agents in formulating its decision functions. Several useful models of coordination and cooperation are thus possible.

As a research group, we have invested a lot of time exploring the basic principles of praxeic utility. In doing this, several example applications have been reviewed, among them the linear quadratic regulator problem, and the inverted pendulum problem (single user problems); an extension of the *laissez faire* problem (a resource allocation problem); the problem of assigning pilots to planes with constraints regarding skill level and pilot satisfaction (negotiated multi agent decisions); “capture the flag,” a dynamic game of pursuit (a coordinated decision problem); and the prisoners’ dilemma, a famous example of a difficult two-person game from game theory (an example of a deliberative solution). The intent for working through these examples was to provide a background for problems of immediate interest, including the “screaming generals” problem that has been presented as a model for task assignment.

Continued investigation of the general praxeic utility theory may lead to interesting research questions. For example, some problems appear to admit only a coordinated solution (each agent taking into account the presence and actions of other agents, but without the explicit requirement that all solutions be collectively acceptable.) However, as the constraints on the problem increase, it may be necessary to enforce group preferences. The behavior in the transition region between coordinated and negotiated decisions seems like an interesting question, since it appears to be on the boundary of where solutions become hardest to find. Another related question is to determine regimes where individual coordinated solutions are superior (in some measure) to negotiated solutions.

There was another aspect of discussion in the research group, motivated in part by issues related to praxeic utility. Formulating the joint selectability rejectability function can be computationally difficult task, especially if the decision spaces and the number of users are large. One of the alternatives we explored is the concept of economic equilibrium, where the “negotiated” solution depends finally on a price vector which establishes a collective equilibrium. One of the intended directions for ongoing research is the possibility of using such economic models to formulate the selectability and rejectability functions, bringing to bear the richness of economic modeling with the flexibility of praxeic utility theory.

4.2 An in-depth look at praxeic decision theory

The following discussion is derived from [17] and [18].

Traditional engineering design methodologies are caught in "the grip of optimality": Virtually every engineering design concept can be expressed as $\dot{X} = 0$. For example, optimal estimation, optimal filtering, optimal detection, optimal equalization, and optimal control all are based on principles of optimality. Why is this? Certainly there is historical justification for optimality-based methods, and often analytically tractable solutions are obtained. Looking from a more rational point of view, however, begin "good enough" should suffice in many circumstances, while being "best" is a bonus. However, there has has been no systematic development of the "good enough," while calculus provides us a measure of optimality.

4.2.1 Modes of rationality

In contemplating these design methodologies, we are moved to identify three different modes of rationality,

- Maximize expected utility: **substantively rational** (the best)
- Follow rules or procedures: **procedurally rational** (what we can come up with; heuristic; ad hoc)
- Expected gains exceed expected losses: **intrinsically rational**.

Regarding this trichotomy, the following observation has been made:

> ... the real accomplishment will come in finding an interesting middle ground between hyperrational behavior and too much dependence on *ad hoc* notions of similarity and strategic expectations. When and if such a middle ground is found, then we may have useful theories with situations in which the rules are somewhat ambiguous. (Kreps, 1990, p. 184)

A mode of rationality is desired that addresses

- Adequacy – When is a solution "good enough"?
- Sociality – How can a group rationality be defined? ("Liberation from maximization may open the door to accommodating group as well as individual interests.")
- Intrinsic – the comparisons for selecting an object depend on the object itself (not on other objects):
- Self-Criticism – How can the quality of the solution be gauged?

Regarded from the point of view of information gathering, as a precursor to a more active control stance, we have the following observation:

> Minimizing the probability of error is not equivalent to avoiding error. Indeed, if an expressed aim of our inquiry is to avoid error, we may comply completely by simply refusing to make any choice at all... Evidently, the decision maker must be willing to incur some risk of error if a meaningful decision is to be made. (Stirling, 2000)

As an example in a decision-theoretic context, consider a digital communication system in which the system output is the set-based answer: the bit is either 1 or 0. This (useless) system is guaranteed to have a probability of error which is always zero: the receiver is never incorrect. However, these decisions are useless: the ultra-cautionary stance has lead to a system that is always correct, and simultaneously never informative.

We thus conceive of two desiderata for a decision-making agent: the desire to obtain new information (knowing the truth) and avoiding error.

These observations are echoed by the statement of the logician Whitehead:

> It is more important that a proposition be interesting than that it be true. This statement is almost a tautology. For the energy of operation of a proposition is an occasion of experience is its interest, and its importance. But of course, a true proposition is more apt to be interesting than a false one. (Whitehead, 1937).

4.2.2 Truth and error valuations

One of the tenets of praxeology is that *set* decisions are permissible. We have a set of propositions (choices) U available to a decision making agent X, we contemplate the Boolean algebra of possible choices.

In the interest of obtaining new information (without necessary regard for its veracity or usefulness), X imposes a valuation on the choices available to it. If a proposition is of **low interest** — it is uninformative — then there is a high value in *rejection*. We quantify this by introducing P_R – the informational value of rejection.

- P_R: $\mathcal{F} \to \mathbb{R}$
- $P_R(\emptyset) = 0$ (no value in rejecting nothing). $P_R(U) = 1$ (normalized)
- Additive structure: $P_R(A_1 \cup A_2) = P_R(A_1) + P_R(A_2)$ if $A_1 \cap A_2 = \emptyset$.

For example, by rejecting $\{u_1\} \in \mathcal{F}$, we "conserve" the informational value $P_R(\{u_1\})$.

Under the assumption that only one choice in U is "correct" (true), then the utility of accepting a set $A \in \mathcal{F}$ is

$$I_A(u) = \begin{cases} 1 & u \in A \\ 0 & u \notin A \end{cases}$$

We can regard $I_A(u)$ as the **error avoidance** utility: the utility of not rejecting the set A if u is true.

If an agent X wants only information value, he should go with P_R; if he wants only truth, he should go with I_A. In the epistemological framework of Levi [19] X employs a convex sum of utilities:

$$\phi(A,u) = \alpha I_A(u) + (1-\alpha)(1 - P_R(A)),$$

for $0 \le \alpha \le 1$. $\phi(A,u)$ is the **epistemic utility function**, measuring the epistemic value of *not* rejecting A when u is true. If $u \notin A$, $\phi(A,u) = (1-\alpha)(1-P_R(A))$. If $u \in A$, $\phi(A,u) = \alpha + (1-\alpha)(1-P_R(A))$.

By a change of variables (which does not change the utility ordering), we can write

$$\varphi(A,u) = I_A(u) - bP_R(A)$$

(where $b = (1-\alpha)/\alpha$), which is an equivalent utility function. We shall call b the **index of boldness**.

4.2.3 Expected utility

X does not know which element is true, and so cannot evaluate $\varphi(A,u)$ exactly. However, we assume X has a probability distribution P_S, the **credal probability** to measure the degree to which it believes the propositions.

On the basis of this probability distribution, we form an average:

$$\overline{\varphi}(A) = \int_U \varphi(A,u) P_s(du) = P_S(A) - bP_R(A).$$

This produces the **expected epistemic utility**. For discrete outcomes:

$$P_S(\{u\}) = p_S(u) \qquad P_R(\{u\}) = p_R(u)$$
$$\overline{\varphi}(A) = \sum_{u \in A} [p_S(u) - bp_R(u)]$$

For continuous outcomes:

$$P_S(A) = \int_A p_s(u)\,du \qquad P_R(A) = \int_A p_R(u)\,du$$
$$\overline{\varphi}(A) = \int_A [p_S(u) - bp_R(u)]\,du$$

Based on this utility, we formulate a **set-valued** decision which maximizes the expected utility:

$$S_b = \arg\max_{A \in \mathcal{F}} \overline{\varphi}(A) = \{u \in U \colon p_S(u) \ge bp_R(u)\}$$

This is the set of decisions for which the truth support is greater than or equal to the informational value of rejection. (Any element in this set is an acceptable answer on the basis of the criteria given.)

Under this decision criterion, there is no compulsion to accept only the "best" — nor even a designation as to what best is. There are only those choices which — based on intrinsic valuations — are worth the risk of accepting. Elements not chosen (that is, those in $U \backslash S_b$) are either not likely to be true, or are not worth the risk of choosing even if true.

4.2.4 Decisions → Actions

Discussion to this point has taken the view point of a decision theory, and may be regarded as a discussion in the branch of philosophy known as epistemology:

> **Epistemology**: the study or theory of the origin, nature, methods, and limits of *knowledge*. (What to believe.)

We now shift our viewpoint from one of decision theory to one of control. In making this transition, we make use of the term **praxeology**, introduced by Stirling, as a "control" parallel to epistemology:

> **Praxeology:** the study of theory of practical activity; the science of efficient action. (How to act.)

In making this transition, we map the notions of "truth" and "error" to concepts applicable to a domain of action: We want more than "success," we want "efficient" success. An agent contemplating action employs the rejectability and selectability of his options, defined as follows:

- **Rejectability**: Options are evaluated with respect to the degree of **resource consumption**.
- **Selectability**: Options are evaluated with respect to the degree that it **accomplishes the objective.**

As in the decision-theoretic case, the agent assigns a selectability measure $p_S(u)$ and a rejectability measure $p_R(u)$. Proceeding along parallel lines, we arrive at the maximum praxeic utility decision rule,

$$S_b = \{u \in U \colon p_S(U) \geq b p_R(u)\}$$

Under this rule, options are selected for which the selectality is not less than the rejectability. We designate this test as the PLRT (praxeic utility likelihood ratio test).

4.2.5 Tie breaking

By the stated criteria, any element in S_b is acceptable. However, when it is necessary to reduce the choices to a single one, one of several tie breakers may be used:

- A satisficing option u_S is **most selectable** if $u_S = \arg\max_{u \in S_b}\{p_S(u)\}$.
- A satisficing option u_R is **least rejectable** if $u_R = \arg\min_{u \in S_b}\{p_R(u)\}$.
- A satisficing option u^* is **maximally discriminating** if $u^* = \arg\max_{u \in S_b}\{p_s(u) - b p_R(u)\}$

A satisficing option u_1 is **more satisficing** than an alternative u_2 if u_1 is either (a) not less selectable and less rejectable than u_2 [i.e., $p_S(u_1) \geq p_S(u_2)$ and $p_R(u_1) < p_R(u_2)$] or (b) not more rejectable than u_2 and more selectable than u_2 [i.e., $p_S(u_1) > p_S(u_2)$ and $p_R(u_1) \leq p_R(u_2)$]

A satisficing option is **arbitrary** if it is chosen at random from S_b.

4.2.6 An example: Nonlinear quadratic regulator

To demonstrate the applicability of the praxeic concepts, we sketch here the application to the nonlinear quadratic regular from controls. [18]. Suppose we have a nonlinear time-varying system described by the discrete-time dynamics

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{x}(t), u(t), t], \qquad t = 0, 1, \ldots, t_f - 1$$

with a performance index

$$J = \mathbf{x}^T(t_f)\tilde{P}\mathbf{x}(t_f) + \sum_{t=0}^{t_f-1} \mathbf{x}^T(t+1)Q(t+1)\mathbf{x}(t+1) + R_u(t)u^2(t)$$

We want to determine an input sequence $u(t), t = 0, 1, \ldots, t_f - 1$ to minimize J: move state to origin and minimize costs along the way.

We will use proximity to final goal $\mathbf{x}^T(t_f)\tilde{P}\mathbf{x}(t_f)$ to determine selectability, and incremental costs $\mathbf{x}^T(t+1)Q\mathbf{x}(t+1) + R_u u^2(t)$ to determine rejectability: actions which move toward the goal will rate with high selectability, while actions which require expensive effort are more rejectable. In the interest of implementability, we will use a receding horizon controller, choosing at each time step the input $u(t)$ that is locally the best. Also, assume control over a bounded interval, $u(t) \in (U_{\min}, U_{\max})$.

To define selectability, we proceed as follows. Pretend that the next step is the final one. Define

$$\Phi(u) = \mathbf{x}^T(t+1)\tilde{P}\mathbf{x}(t+1).$$

Smaller distance is better, so we flip this around:

$$g_S(u; \mathbf{x}(t)) = \sup_{v \in (U_{\min}, U_{\max})} \Phi(v) - \Phi(v) + \epsilon$$

Now normalize so it looks like a probability:

$$p_S(u) = \frac{g_S(u; \mathbf{x}(t))}{\int_{U_{\min}}^{U_{\max}} g_S(v, \mathbf{x}(t))\, dv}$$

So: inputs which move us closer to target have higher selectability.

Rejectability is based on incremental costs. Define

$$\Lambda(u) = \mathbf{x}^T(t+1)Q(t+1)\mathbf{x}^T(t+1) + R_u(t+1)u^2(t).$$

Smaller cost is less rejectable: oriented correctly. Shift:

$$g_R(u, \mathbf{x}(t)) = \Lambda(u) - \inf_{v \in (U_{\min}, U_{\max})} \Lambda(v) + \epsilon$$

and normalize:

$$p_R(u, \mathbf{x}(t)) = \frac{g_R(u, \mathbf{x}(t))}{\int_{U_{\min}}^{U_{\max}} g_R(v; \mathbf{x}(t))\, dv}$$

The point of this demonstration at juncture is this: to obtain the selectability and rejectability measures, Simply identify the goals and costs, and quantify them, normalizing as probabilities.

Now let us consider specialization to a linear quadratic regular of a linear system, so that comparisons may be made with conventional optimal control theory. Our system has the model

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + Bu_k, \qquad k = 0, 1, 2, \ldots, t_f.$$

The control goal is to select a control input sequence $\{u_0, u_1, \ldots, u_{t-1}\}$ to drive the state $\mathbf{x}_k$ from arbitrary initial conditions to $\mathbf{x}_t = \mathbf{0}$ in a way to conserve energy in the control. There are two roughly opposing desiderata: Reduce the error at the terminal time $e_1 = \mathbf{x}_{t_f}^T \mathbf{x}_{t_f}$, and to use as little energy as possible, $e_2 = \sum_{k=0}^{t_f-1} u_k^2$. In conventional optimal control, these are combined into a weighted sum:

$$J(u_0, \ldots, u_{t_f-1}) = \mathbf{x}_{t_f}^T \mathbf{x}_{t_f} + \sum_{k=0}^{t_f-1} Ru_k^2.$$

Linear optimal control provides a well-known answer. By classical optimal control, $u_k = -K_k \mathbf{x}_k$, where K_k is the Kalman gain, $K_k = [B^T P_{k+1} B + R]^{-1} B^T P_{k+1} A$, with $P_k = A^T P_{k+1}(A - BK_k)$.

Regarding this optimal solution we may make the following observations:

- The "optimal" may not really be needed (it is simply suggested as a way to formulate the problem).
- Choosing a single performance index J is arbitrary, as is the weighting between the desiderata measures.
- There may be different units in e_1 and e_2; in some sense they are incommensurables.

In the praxeic approach to this problem, we do not seek a global optimum over the entire ensemble. We consider, therefore, a receding horizon of d control inputs $\{u_k, \ldots, u_{k+d-1}\}$ computed as a function of state $\mathbf{x}_k$. The control u_k is implemented, moving to state $\mathbf{x}_{k+1}$, and the process is repeated. We will deal with $d = 1$ (one-step look-ahead).

We begin by imposing upper and lower bounds on the control variable: $U_m = (-u_m, u_m)$.

The selectability, associated with the target position goal, is defined as follows: If $k + 1$ were the terminal time, the cost associated with this time would be $\Phi(u, \mathbf{x}_k) = \mathbf{x}_{k+1}^T \mathbf{x}_{k+1} = [A\mathbf{x}_k + Bu]^T [A\mathbf{x}_k + Bu]$. (If this were our only constraint, we could move immediately to the correct final value). We define the function

$$g_S(u, \mathbf{x}_k) = \sup_{v \in U_m} \{\Phi(v, \mathbf{x}_k) - \Phi(u, \mathbf{x}_k)\}$$

We want control values that make Φ small to have high selectivity. (We will normalize momentarily).

The rejectability, associated with control costs, is simply taken to be proportional to power: $g_R(u) = Ru^2$.

We can restrict the range to an region (these can be computed because they are quadratic functions)

$$u_{\mathfrak{C}_S} = \text{arg}\max_{u \in U_m} g_S(u, \mathbf{x}_k) \qquad u_{\mathfrak{C}_R} = \text{arg}\max_{u \in U_m} g_R(u)$$

Then we define the equilibrium set $u_* = \min(u_{\mathfrak{C}_S}, u_{\mathfrak{C}_R})$ $u^* = \max(u_{\mathfrak{C}_S}, u_{\mathfrak{C}_R})$, $U = [u_*, u^*]$.

Now let

$$p_S(u; \mathbf{x}_k) = \frac{g_S(u, \mathbf{x}_k)}{G_S(\mathbf{x}_k)} \qquad p_R(u) = \frac{g_R(u)}{G_R(\mathbf{x}_k)}$$

where

$$G_S(\mathbf{x}_k) = \int_{u_*}^{u^*} g_S(v; \mathbf{x}_k)\, dv \qquad G_R(\mathbf{x}_k) = \int_{u_*}^{u^*} g_R(v)\, dv$$

are normalizing terms.

Most selectable: $u_S = -[B^T B]^{-1} B^T A \mathbf{x}_k$. Least rejectable: $u_R = 0$.

Observe that $p_S(u, \mathbf{x}_k)$ is concave and $p_R(u)$ is convex. Thus all possible satisficing equilibrium controls are obtained as convex combinations of u_R and u_S: $u_\lambda = \lambda u_R + (1 - \lambda) u_S$.

Most discriminating control is that which maximizes $p_S(u; \mathbf{x}_k) - b p_R(u)$ with respect to u. This gives

$$u_k = -[B^T B + b' R]^{-1} B^T A \mathbf{x}_k,$$

where $b' = b\frac{G_S(\mathbf{x}_k)}{G_R(\mathbf{x}_k)}$. We thus have $u_k = -\mathcal{K}_k \mathbf{x}_k$, where $\mathcal{K}_k = [B^T B + b' R]^{-1} B^T A$, which is state-feedback (like optimal), but not linear since b' depends on $\mathbf{x}_k$.

An example of this control law is as follows:

$$A = \begin{bmatrix} 0.9974 & 0.0539 \\ -0.1078 & 1.1591 \end{bmatrix} \qquad B = \begin{bmatrix} 0.0013 \\ 0.0539 \end{bmatrix} \qquad R = 0.05.$$

Then the control input $u[k]$ is shown in figure 4.1, and the phase-plane trajectories are shown in figure 4.2.6, where the solid line indicates the optimal solution, and the dashed lines indicate the praxeic control for $b = 1$ and $b = 1.6$.

4.3 Extending praxeic utility to multi-agent systems

As before, this material is drawn closely from [20].

We have seen that the praxeic viewpoint eschews the imperative to select only the best solution, considering in addition solutions which are arguably "good enough." This additional flexibility is important when dealing with multiagent systems. In this section we discuss aspects of this multiagent application.

As Stirling has observed,

> "Group rationality is not a logical consequence of rationality based on individual self interest. Under substantive rationality, where maximization of satisfaction is the operative norm, group behavior, consisting of the collection of individual behaviors, is not

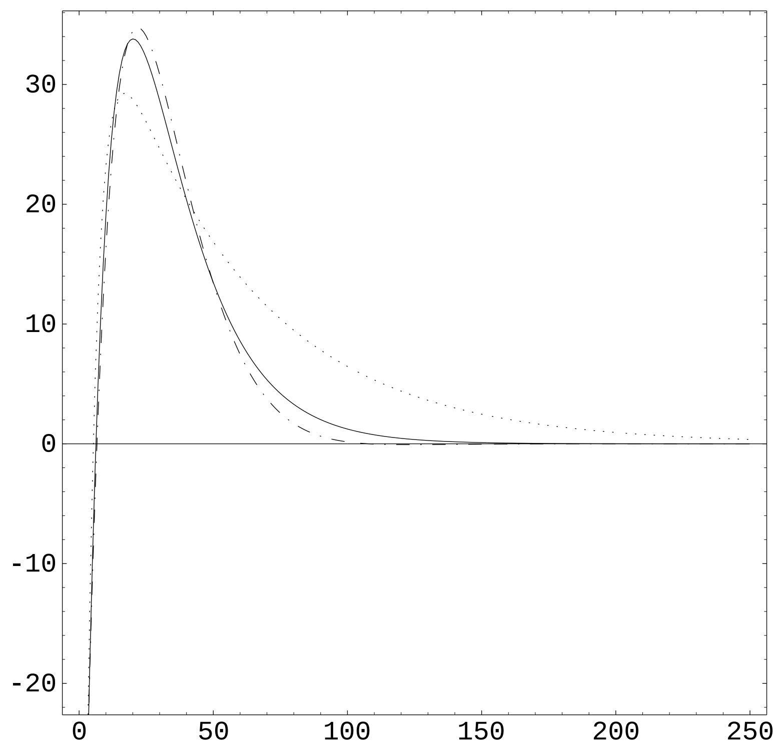


Figure 4.1: Control input for the linear quadratic regulator

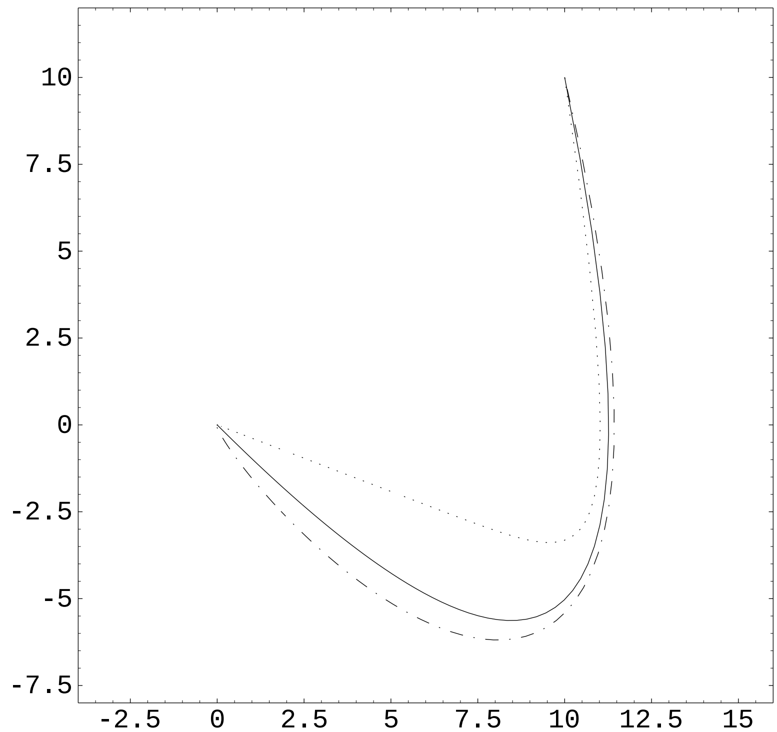


Figure 4.2: Phase trajectory for linear quadratic regulator

usually optimized by optimizing each individual behavior, as is done under conventional game theory. Unfortunately, those who put their final confidence in the limited perspective of exclusive self-interest may ultimately function disjunctively, and perhaps illogically, when participating in collective inferences."

Coordination among agents is a neutral concept: Cooperation, negotiation, competition, and any other form of behavior short of complete indifference and isolation between agents will involve some form of coordination. Any decision by an agent that uses information concerning the existence, decisions, or decision-making strategies of any other agent is a coordinated decision.

4.3.1 The View from the Praxeic Utilitarian

1. My understanding (or estimate) of your selectability and rejectability functions may affect my own selectability and rejectability.

2. Group decisions require the formulation of *joint* selectability and rejectability.

3. Set decisions provide more flexibility for achieving jointly acceptable solutions.

4. Boldness becomes a tool for negotiation.

In the praxeic framework, a single optimum strategy is not sought. Instead, each agent considers options based on joint selectability and rejectabilities, where a selectability for an agent may incorporate selectabilities and/or rejectabilities of other agents. These utility functions have the properties of probabilities and may frequently be codified using conditional probabilities, which essentially provides an interpolated rule-set for the agents. This framework can encompass traditional game theory, but provides in addition much more flexibility and a closer approximation to the operations of human deliberations.

4.3.2 Notation

In the problem formulation each agent $X_1, X_2, \ldots, X_N$, is endowed with its own decision space U_i, and joint decision space is formed as the cross product of these decision spaces. $U_{1:N} = U_1 \times \cdots \times U_N$. There is also a joint Boolean algebra over these decision spaces, $\mathcal{F}_{1:N} = \mathcal{F}_1 \times \cdots \times \mathcal{F}_N$.

The joint rejectability function $m_{1:N} : \mathcal{F}_{1:N} \to [0,1]$ maps a joint decision to a rejectability value. The joint selectability function: $Q_{1:N} : \mathcal{F}_{1:N} \to [0,1]$ maps a joint decision to a selectability value.

Each agent has its own boldness: b_i, and a joint boldness is computed as a product: $b_{1:N} = b_1 b_2 \cdots b_N$.

The selectabilities and rejectabilities of all agents are represented in a coordination function: A joint inter-agent rejectability and credibility function:

$$f_{R_1 S_1 R_2 S_2 \ldots R_N S_N}(x^1, y^1, x^2, x^2, \ldots, x^N, y^N)$$

for $(x^1, \ldots, x^N) \in U_{1:N}$ and $(y^1, \ldots, y^N) \in U_{1:N}$. (This illustrates a potential serious problem with this approach: the argument space is very large.)

Once a joint inter-agent rejectability and credibility function is computed, the selectability and rejectability for individual agents can be computed as marginals of the coordination function.

Frequently, the inter-agent rejectability and credibility function is obtained by conditioning; this can reduce the effective dimensionality of the problem. For example,

$$f_{R_1,S_1,R_2,S_2} = f_{R_1,S_1|R_2,S_2} f_{R_2,S_2}$$

$f_{R_1,S_1|R_2,S_2}$ represents X_1's values (expressed as rejectability) and beliefs (expressed as selectability), conditioned upon what X_2 values and believes. Note that $f_{R_2,S_2} = f_{R_2} f_{S_2}$.

Further conditioning: $f_{R_1,S_1|R_2,S_2} = f_{S_1|R_1,R_2,S_2} f_{R_1|R_2,S_2}$. We will set up some parameterized values for these functions.

4.3.3 An illustrative example: The prisoner's dilemma

The prisoner's dilemma is a famous problem in game theory; we present it here to illustrate how the praxeic concepts can be extended to multiple agent systems. In this game, two agents, X_1 and X_2, have been charged with a serious crime, arrested, and incarcerated so they cannot communicate. Prosecution has evidence to convict of a lesser offense. To get at least one conviction on the more serious crime, the prosecution entices each prisoner to give evidence against the other in return for dropping charges. If both confess, each receives a prison term of intermediate length (for cooperation). The payoff matrix for the game follows.

	X_2	
X_1	silent	confesses
silent	2,2	4,1
confesses	1,4	3,3

The choices in this game will be denoted by H_0: silence, H_1: confess. (Denote choice by 0 or 1). Then a reasonable assignment to a factored joint function, in terms of the rejectability, might be:

- $f_{R_1|R_2,S_2}(0|0,0) = 1$, $f_{R_1|R_2,S_2}(1|0,0) = 0$: Given X_2 values rejecting silence (confession), rejecting silence is preferred. Given X_2 believes in silence, it would be in the interest to be silent. By this conditioning, X_2 is confused, so we go with the safe option.

- $f_{R_1|R_2,S_2}(0|1,0) = 1-\epsilon$, $f_{R_1|R_2,S_2}(1|1,0) = \epsilon$: X_2 values and believes in silence. $1-\epsilon$ is the degree to which X_1 wants to *exploit* this willingness.

- $f_{R_1|R_2,S_2}(0|0,1) = 1-\mu$, $f_{R_1|R_2,S_2}(1|0,1) = \mu$: X_2 values and belies in confession. μ is the the degree to which X_1 is willing to be exploited (martyrdom).

- $f_{R_1|R_2,S_2}(0|1,1) = 1$, $f_{R_1|R_2,S_2}(1|1,1) = 0$: again, X_2 is confused, and we go with the safe option.

The parameters μ and ϵ represent willingness to give and take, but it is not necessary that $\mu = 1 - \epsilon$.

The selectability function can be represented as

- $f_{S_1|R_1,R_2,S_2}(0|0,0,0) = 0$, $f_{S_1|R_1,R_2,S_2}(1|0,0,0) = 1$, $f_{S_1|R_1,R_2,S_2}(0|1,0,0) = 0$, $f_{S_1|R_1,R_2,S_2}(1|1,0,0) = 1$: X_2 is confused: go with the safe bet.

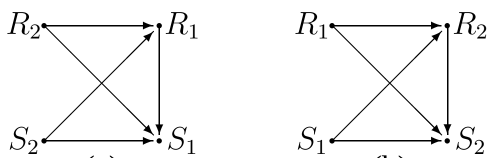
- $f_{S_1|R_1,R_2,S_2}(0|0,0,1) = \sigma$, $f_{S_1|R_1,R_2,S_2}(1|0,0,1) = 1 - \sigma$: X_2 wants to confess. σ measures X_1's belief that best interest lies in self defense.

- $f_{S_1|R_1,R_2,S_2}(0|1,0,1) = \chi$, $f_{S_1|R_1,R_2,S_2}(1|1,0,1) = 1 - \chi$: χ measures masochism: acting against values.

- $f_{S_1|R_1,R_2,S_2}(0|0,1,1) = 0$, $f_{S_1|R_1,R_2,S_2}(1|0,1,1) = 0$, $f_{S_1|R_1,R_2,S_2}(0|1,1,1) = 0$, $f_{S_1|R_1,R_2,S_2}(1|1,1,1) = 0$: X_2 is confused: go with the safe bet.

4.3.4 Agent reasoning and deliberation

In addition to "static" representations of rejectability and selectability, there is also the notion of reaction: A model which enables X_1 to predict X_2's decision is said to be *reactive*. A reactive model is illustrated here:



(a) (b)

In the context of reactive response, or deliberation, let $\hat{f}_{R_2S_2|R_1S_1}$ be the estimate X_1 has of X_2's condition coordination function, and let $\hat{f}^{[0]}_{R_2S_2}$ be X_1's *a priori* estimate of X_2. X_1 may obtain an initial estimate of the coordination function as

$$f^{[1]}_{R_1S_1R_2S_2} = f_{R_1S_1|R_2S_2} f^{[0]}_{R_2S_2}.$$

Then the joint rejectability/selectability is

$$f^{[1]}_{R_1S_1}(r^1, s^1) = \sum_{r^2 \in U_2} \sum_{s^2 \in U_2} f^{[1]}_{R_1S_1R_2S_2}(r^1, s^1, r^2, s^2)$$

- X_1 may adopt a reactive decision now: form

  $$f^{[1]}_{R_1}(r^1) = \sum_{s^1 \in U^1} f^{[1]}_{R_1S_1}(r^1, s^1)$$

  and

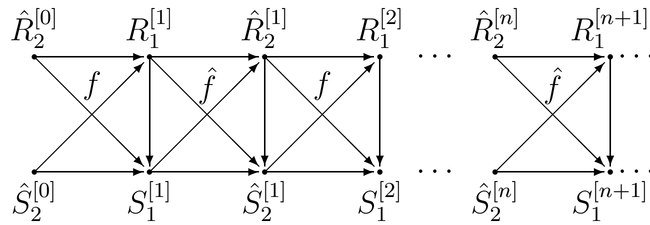
  $$f^{[1]}_{S_1}(s^1) = \sum_{r^1 \in U^1} f^{[1]}_{R_1S_1}(r^1, s^1),$$

  then use the rule of epistemic utility.

- **Or** X_1 may adopt a deliberative strategy: X_1 may impugn to X_2 his own methods: $\hat{f}^{[1]}_{R_1 S_1 R_2 S_2} = \hat{f}_{R_2 S_2 | R_1 S_1} f^{[1]}_{R_1 S_1}$,

$$\hat{f}^{[1]}_{R_2 S_2}(r^2, s^2) = \sum_{s^1 \in U_1} \sum_{r^1 \in U_1} \hat{f}^{[1]}_{R_1 S_1 R_2 S_2}(r^1, s^1, r^2, s^2)$$

An agent can “deliberate” many iterations.



$$f^{[n+1]}_{R_1 S_1 R_2 S_2} = f_{R_1 S_1 | R_2 S_2} \hat{f}^{[n]}_{R_2 S_2}$$

$$f^{[n+1]}_{R_1 S_1}(r^1, s^1) = \sum_{s^2 \in U_2} \sum_{r^2 \in U_2} f^{[n+1]}_{R_1 S_1 R_2 S_2}(r^1, s^1, r^2, s^2)$$

$$\hat{f}^{[n+1]}_{R_1 S_1 R_2 S_2} = \hat{f}_{R_2 S_2 | R_1 S_1} f^{[n+1]}_{R_1 S_1}$$

$$\hat{f}^{[n+1]}_{R_2 S_2}(r^2, s^2) = \sum_{s^1 \in U_1} \sum_{r^1 \in U_1} \hat{f}^{[n+1]}_{R_1 S_1 R_2 S_2}(r^1, s^1, r^2, s^2).$$

The deliberation can be expressed in terms of matrices, and the convergent solution can be found: the eigenvector corresponding to the unit eigenvalue of the appropriate matrix.

Lemma 1 *When $b = 1$, cooperation cannot occur unless $\chi > 0$ and $\mu > 0$.*

For example, in the Prisoner’s dilemma, when $\chi \to 0$, we find the rejectability and selectability of the fixed point to be

$$\mathbf{m}_1^* = \begin{bmatrix} \dfrac{1+\sigma-\mu\sigma}{1+\mu+\sigma-\mu\sigma} \\ \dfrac{\mu}{1+\mu+\sigma-\mu\sigma} \end{bmatrix} \qquad \mathbf{q}_1^* = \begin{bmatrix} \dfrac{\sigma-\mu\sigma}{1+\mu+\sigma-\mu\sigma} \\ \dfrac{1+\mu}{1+\mu+\sigma-\mu\sigma} \end{bmatrix}$$

So if there is any hope of cooperation, each prisoner must to some degree value martyrdom ($\mu > 0$) and must also have non-zero selectability that a masochistic decision is in its best interest ($\chi > 0$).

Lemma 2 *When $b = 1$ and $\chi, \mu > 0$, silence (H_0) is selected when $\chi\mu \geq e + \epsilon - \epsilon e$.*

Note that this does not depend upon c or σ.

4.3.5 General formulation of multiagent epistemic/praxeic decision making

In the general case, we form a joint selectability/rejectability function

$$p_{S_1,S_2,\ldots,S_N,R_1,R_2,\ldots,R_N}(u_1,u_2,\ldots,u_N,v_1,v_2,\ldots,v_N) = p_{\mathbf{SR}}(\mathbf{u},\mathbf{v}),$$

where $u_i \in U_i$ and $v_i \in U_i$. Let $\mathbf{U} = U_1 \times U_2 \times \cdots \times U_N$.

As we have seen, determining the values is frequently accomplished by factorizations. For example in a two-agent system we might have

$$\begin{aligned} p_{S_1 S_2 R_1 R_2}(u_1,u_2,v_1,v_2) &= p_{S_1|S_2 R_1 R_2}(u_1|u_2,v_1,v_2)p_{S_2|R_1 R_2}(u_2|v_1,v_2) \\ &\qquad p_{R_1|R_2}(v_1|v_2)p_{R_2}(v_2) \\ &= p_{S_1|S_2 R_2}(u_1|u_2,v_2)p_{S_2|R_1}(u_2|v_1)p_{R_1|R_2}(v_1|v_2)p_{R_2}(v_2) \\ &\qquad \text{(selectability does not depend on rejectability)} \end{aligned}$$

We form the **multipartite selectability** and the **multipartite rejectability** as marginals:

$$p_{\mathbf{S}}(\mathbf{u}) = \sum_{\mathbf{v}\in\mathbf{U}} p_{\mathbf{SR}}(\mathbf{u},\mathbf{v}) \qquad\qquad p_{\mathbf{R}}(\mathbf{v}) = \sum_{\mathbf{u}\in\mathbf{U}} p_{\mathbf{SR}}(\mathbf{u},\mathbf{v})$$

The **Multipartite Decision rule** forms the multipartite satisficing set as

$$\mathbf{S}_b = \{\mathbf{u} \in \mathbf{U}\colon p_{\mathbf{S}}(\mathbf{u}) \geq b p_{\mathbf{R}}(\mathbf{u})\}.$$

Individual selectability and rejectabilities are computed using further marginalization:

$$p_{S_i}(u_i) = \sum_{1\leq j\leq N, j\neq i} p_{S_1,\ldots,S_N}(u_1,\ldots,u_N) \qquad\qquad p_{R_i}(v_i) = \sum_{1\leq j\leq N, j\neq i} p_{R_1,\ldots,R_N}(v_1,\ldots,v_N)$$

Then an **Individual Satisficing set** is formed as

$$S_b^i = \{u_i \in U_i : p_{S_i}(u_i) \geq b p_{R_i}(u_i)\}$$

Denoting $\mathbf{R}_b = S_b^1 \times S_b^2 \times \cdots \times S_b^N$ as the **satisficing rectangle** for all agents, we may ask: is $\mathbf{R}_b = \mathbf{S}_b$? Is the collection of individual decisions equivalent to the multipartite decision? That is, do the individual decisions coincide with the group decisions? In general, the answer is no. In this case, there is need for negotiation.

4.3.6 Negotiation

We may make the following observations about negotiation:

- Negotiators are usually more concerned with meeting minimal requirements than with achieving maximum performance.
- Negotiations should lead to decisions that are good enough for the group as a whole and good enough for each individual.

- Negotiation is a narrowing down of options.

A model for multiagent decision making should reasonably support a method of negotiation which supports these concepts. As we now show, this is the case,

The set $\mathbf{S}_b$ (the jointly satisficing set) represents choices that are good enough for the group. The set $\mathbf{R}_b$ (the satisficing rectangle) represents choices that are good enough for the individuals in the group.

The **Negotiation Theorem** states: It can be shown that if s_i is individually satisficing for X_i, that is $s_i \in S_b^i$, then it must be a corresponding element in some jointly satisficing vector $\mathbf{s} \in \mathbf{S}_b$.

By this theorem, no one is "frozen out" of a deal.

In the context of satisficing, a means of representing the "lowering of standards" for group accommodation is the boldness.

Let b_i be the boldness for X_i, $\mathbf{b} = (b_1, \ldots, b_N)$, and $b_L = \min\{b_1, \ldots, b_n\}$.

Now form a compromise set of choices that are *individually satisficing*:

$$\mathbf{C}_i = \{\mathbf{s} = \{s_1, \ldots, s_N\} \in \mathbf{S}_{b_L} : s_i \in S_{b_i}\}$$

(Uses b_L: the standards of a group can be no higher than the standards of any member of a group.)

A choice $\mathbf{s} = (s_1, \ldots, s_N)$ is a satisficing imputation at boldness $\mathbf{b}$ if $p_{\mathbf{S}}(\mathbf{s}) \geq b_L p_{\mathbf{R}}(\mathbf{s})$ and $p_{S_i}(s_i) \geq b_i p_{R_i}(s_i)$ for $i = 1, 2, \ldots, N$: it is jointly satisficing for the group, and each component is individually satisficing for its corresponding member of the group. The **satisficing imputation set** $\mathbf{N}_b$ is the set of satisficing imputations:

$$\mathbf{N}_b = \cap_{i=1}^{N} \mathbf{C}_i.$$

A method of negotiation is to have each agent lower its own boldness until a non-empty $\mathbf{N}_b$ is obtained.

Based on this concept, we present the following algorithm for negotiation:

Each agent chooses a boldness b_i (typically $b_i = 1$ to start)

1. X_i forms $\mathbf{S}_{b_L}^i$ and $S_{b_i}^i$ for $i = 1, 2 \ldots, N$.
2. X_i forms its compromise set by eliminating all choices for which its component is not individually satisficing. This gives $\mathbf{C}_i = \{s \in \mathbf{S}_{b_L}^i \colon s_i \in S_{b_i}^2\}$.
3. X_i shares $\mathbf{C}_i$ and b_i to all other agents.
4. The satisficing imputation set $\mathbf{N}_\mathbf{b} = \cap_{j=1}^{N} \mathbf{C}_j$ is formed. If $\mathbf{N}_b bf = \emptyset$, then decrement b_j for $j = 1, \ldots, N$, and return to step 1.
5. After completion, X_i implements the ith component of the rational compromise

$$\mathbf{s}^* = \arg \max_{\mathbf{s} \in \mathbf{N_b}} \frac{p_{S_1, \ldots, S_N}(\mathbf{s})}{p_{R_1, \ldots, R_N}(\mathbf{s})}.$$

4.3.7 A simple example of negotiation

N Pilots $X_1, \ldots, X_N$ to collectively fly $M < N$ aircraft for mission k. Let $I(k) = \{i_1, \ldots, i_M\}$ denote the set of indices of participants, $1 \le i_j \le N$. Each X_i has a skill level $s_i(k)$. Let $\mathbf{s}(k) = \{s_1(k), \ldots, s_N(k)\}$. Let $\boldsymbol{\sigma}(k) = \{s_{i_1}(k), \ldots, s_{i_M}(k)\}$, $i_j \in I(k)$ be the skill levels of the participants on mission k. Let $g_i(s)$ denote the flyer's satisfaction, with g_i nondecreasing in s, and $0 \le g_i(s) \le 1$. Skill increase with experience:

$$s_i(k+1) = \begin{cases} f_1[g(\boldsymbol{\sigma}(k), s_i(k)] & \text{if } i \in I(k) \\ f_2[s_i(k) & \text{if } i \notin I(k). \end{cases}$$

Let $g[\boldsymbol{\sigma}(k)]$ denote the joint satisfaction function for the group. Each mission incurs risk to fliers: we assume risk function depends on *least-skilled* flyer in group $r(\min_{i \in I(k)} s_i(k))$. The Individual agents' Goal: to increase skill level (satisfaction). The Group Goal: all participants to increase skill levels, perhaps uniformly.

Agents must negotiate to obtain group decision.

Let $U_i = \{1, 0\}$, indicating `fly` or `don't fly`. Group decision: $\mathbf{U} = \{0,1\}^N$. The decision vector (of length N) must have exactly M 1s in it; there are $\binom{N}{M}$ possible choices in this set, which we designate as $\mathbf{U}_N$.

For a $\mathbf{u} \in \mathbf{U}_N$, we can write the skill vector $\boldsymbol{\sigma}(k)$ as $\boldsymbol{\sigma}(k) = \Phi(\mathbf{u}(k))\mathbf{s}(k)$, where $\Phi(\mathbf{u})$ maps the vector to a matrix, as

$$\Phi(1100) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

We also have $g[\boldsymbol{\sigma}(k)] = g[\Phi(\mathbf{u}(k))\mathbf{s}(k)]$.

Selectability: (Want to maximize collective skills)

$$p_{\mathbf{S}}(\mathbf{u}; \mathbf{s}(k)) = \begin{cases} \frac{g(\Phi(\mathbf{u})\mathbf{s}(k))}{\sum_{\mathbf{v} \in \mathbf{U}_N} g[\Phi(\mathbf{v})\mathbf{s}(k)]} & \mathbf{u} \in \mathbf{U}_N \\ 0 & \mathbf{u} \notin \mathbf{U}_N. \end{cases}$$

Rejectability: (Minimize risk)

$$p_{\mathbf{R}}(\mathbf{u}; \mathbf{s}(k)) = \begin{cases} \frac{r(\Phi(\mathbf{u})\mathbf{s}(k))}{\sum_{\mathbf{v} \in \mathbf{U}_N} \min r(\Phi(\mathbf{v})\mathbf{s}(k))} & \mathbf{u} \in \mathbf{U}_N \\ \infty & \mathbf{u} \in \mathbf{U}_N. \end{cases}$$

Skill update: Let $f_1[g(\Phi(\mathbf{u}(k))\mathbf{s}(k)), s_i(k)] = As_i(k) + \alpha(g(\Phi(\mathbf{u}(k))\mathbf{s}(k)))\Delta s$, where $0 < A < 1$ and

$$\alpha(x) = \frac{1}{1 + e^{-a(x-x_0)}}$$

Let $f_2(s_i(k)) = As_i(k)$ (atrophy without use).

Satisfaction: Let $g_i(s) = \frac{1}{1+e^{-a(s-s_0)}}$.

Risk: Take $r(x) = \frac{1}{2} + \frac{1}{2}e^{-hx}$ for some $h > 0$.

As a particular example, assume $N = 4, M = 2$.

The group selectability can be factored as

$$P_{S_1S_2S_3S_4}(u_1, u_2, u_3, u_4) = P_{S_1|S_2S_3S_4}(u_1|u_2, u_3, u_4)P_{S_2|S_3,S_4}(u_2|u_3, u_4)P_{S_3}(u_3)P_{S_4}(u_4)$$

(E.g., assume that X_3's preferences are independent of X_4's preferences.)

Now we need to specify all the conditional selectabilities:

$$p_{S_3}(1; \mathbf{s}(k)) = 1 - g_3(s_3(k)) \qquad p_{S_3}(0; \mathbf{s}(k)) = g_3(s_3(k)),$$

similarly for p_{S_4}.

For $p_{S_2|S_3,S_4}$: If both X_3 and X_4 ascribe their entire preference to flying, X_2 should not elect to fly. Otherwise, X_2 should go with its individual preferences.

$$p_{S_2|S_3,S_4}(1|1, 1; \mathbf{s}(k)) = 0 \qquad p_{S_2|S_3,S_4}(0|1, 1; \mathbf{s}(k)) = 1$$

$$p_{S_2|S_3,S_4}(1|1, 0; \mathbf{s}(k)) = 1 - g_2(s_2(k)) \qquad p_{S_2|S_3,S_4}(0|1, 0; \mathbf{s}(k)) = g_2(s_2(k)),$$

and so forth.

For $p_{S_1|S_2,S_3,S_4}$: If any two conditioning agents place their entire unit of preference on flying, then X_1 should not elect to fly. Otherwise, X_1 should go with its myopic preferences. Putting these all together we obtain the group preference function:

$$\begin{aligned}
P_{S_1S_2S_3S_4}(1, 1, 0, 0; \mathbf{s}(k)) &= (1 - g_1(s_1(k)))(1 - g_2(s_2(k)))g_3(s_3(k))g_4(s_4(k)) \\
P_{S_1S_2S_3S_4}(1, 0, 1, 0; \mathbf{s}(k)) &= (1 - g_1(s_1(k)))g_2(s_2(k))(1 - g_3(s_3(k)))g_4(s_4(k)) \\
P_{S_1S_2S_3S_4}(1, 0, 0, 1; \mathbf{s}(k)) &= (1 - g_1(s_1(k)))g_2(s_2(k))g_3(s_3(k))(1 - g_4(s_4(k))) \\
P_{S_1S_2S_3S_4}(0, 1, 1, 0; \mathbf{s}(k)) &= g_1(s_1(k))(1 - g_2(s_2(k)))(1 - g_3(s_3(k)))g_4(s_4(k)) \\
P_{S_1S_2S_3S_4}(0, 1, 0, 1; \mathbf{s}(k)) &= g_1(s_1(k))(1 - g_2(s_2(k)))g_3(s_3(k))(1 - g_4(s_4(k))) \\
P_{S_1S_2S_3S_4}(0, 0, 1, 1; \mathbf{s}(k)) &= g_1(s_1(k))g_2(s_2(k))(1 - g_3(s_3(k)))(1 - g_4(s_4(k)))
\end{aligned}$$

After all this (to demonstrate the mechanics of the mathematics), the results can be summarized (after simulation): With arbitrary initial conditions for skill levels, simulations converge to equal skills.

4.4 An example application: Resource Allocation

A system of N agents $\{X_1, \ldots, X_N\}$ are to be assigned to do a number of tasks. For the sake of definiteness, we will assume three distinct classes of tasks (from which generalization should be clear). We will refer to the tasks as flying, sailing, and swabbing — tasks that require vastly different skill sets. Each agent is endowed with a skill set that increases with use on a particular task. There are also physical resources necessary for tasks, although not enough that everyone can be assigned to the resource for every mission. These resources are planes, boats, and mops. There is a desire among agents to increase skill in either flying or sailing (depending upon the classification of the agent), but probably no desire to improve skill at swabbing.

In addition, there is another agent X_0 representing “the management,” whose job it is to see that the mandatory tasks are completed. Management is only incidentally interested in seeing to the general satisfaction of the others.

The jobs at time t can be decomposed into a class of “mandatory jobs and a class of “optional jobs.” Mandatory jobs are those that, from the perspective of management, must be done — actual missions to accomplish something. The optional jobs are available for workers (who, for example, might wish to increase their skills), but are not first priority. Actual missions may involve higher risk than other missions, or greater discomfort. Management always wants to see the mandatory jobs done first; the other agents might prefer the optional jobs first due to their lower risk. (We will assume that in either case the skill level increase is the same for both mandatory or optional jobs; down the road we might want to consider varying these, to build in the fact that there should be better rewards for riskier duty.)

We denote by $M^j(t)$ the requested number of tasks to be performed of type j at time t, and write $M^1(t) + M^2(t) + M^3(t) = M(t)$. Of the $M^j(t)$ jobs of type j, we will denote $\overline{M}^j(t) \leq M^j(t)$ of them as mandatory. The number of agents actually assigned to task j is $\check{M}^j(t)$, and it will always be the case that $\check{M}^j(t) \leq M^j(t)$. (Can’t fill more missions than there is room for.) We take $\check{M}(t) = \check{M}^1(t) + \check{M}^2(t) + \check{M}^3(t)$.

(In a more complete model, aspects of the mission related to piloting skills, such as time of day, light levels, etc., should be incorporated.)

The decision space for each X_i is the set $\{$*do nothing*, *fly*, *sail*, *swab*$\}$, which we represent as $U_i = \{0, 1, 2, 3\}$. The group decision space is $\mathbf{U} = \{0, 1, 2, 3\}^N$. The decision vector $\mathbf{u}(t)$ is an integer string of length N, such that $d_H(\mathbf{u}(t), \mathbf{0}) = \check{M}(t)$. ($d_H$ is the Hamming distance.) Then $[\mathbf{u}(t)]_i$ is the task assigned to agent X_i. $\check{M}^j(t)$ is a function of the decision vector at time t:

$$\check{M}^j(t) = \check{M}(\mathbf{u}, t) = |\{\mathbf{u}_i = j, i = 1, 2, \ldots, N\}|,$$

where $|A|$ denotes the number of elements in the set A.

The set of admissible combinations of choices $\mathcal{U}(t)$ is determined by

$$\mathcal{U}(t) = \{\mathbf{u} \in \mathbf{U}(t) : |\check{M}^j(\mathbf{u}) \leq M_j(t), j = 1, 2, 3\}$$

This allows for the possibility of incomplete missions.[1]

4.4.1 Agent descriptions

1. For an agent $X_i, i > 0$, let its skill vector at time t be

$$\mathbf{s}_i(t) = \left[s_i^1(t), s_i^2(t), s_i^3(t)\right]$$

   where the superscript indicates the task, $1 =$ flying, $2 =$ sailing, and $3 =$ swabbing.

[1] A key issue to be developed is localization. The computational complexity of this problem is going to grow very rapidly. Can the overall computations be partitioned into reasonable subproblems?

Another key aspect that should be examined is the revision of schedules: what if a schedule is established, and new conditions develop. Can a means be found to modify the schedule (with minimum impact while still finding an effective schedule).

2. The skill matrix across all agents is

$$S(t) = \begin{bmatrix} \mathbf{s}_1(t) \\ \vdots \\ \mathbf{s}_N(t) \end{bmatrix}.$$

3. Let $I^1(t)$ indicate the set of indices of participants on the flying mission, $I^1(t) = \{i_1^1, i_2^1, \ldots, i_{\check{M}^1}^1\}$, where $\check{M}^1 = \check{M}^1(t)$ is the number of agents on the first mission at time t. Similarly define $I^2(t)$ and $I^3(t)$ for sailing and swabbing, respectively. Note that $I^i \cap I^j = \emptyset$ for $i \neq j$, and that $\check{M}^1(t) + \check{M}^2(t) + \check{M}^3(t) = \check{M}(t)$. The number of missions of each type may vary at each time.

4. Let

$$\boldsymbol{\sigma}^1(t) = \{s_{i_1^1}^1(t), s_{i_2^1}^1(t), \ldots, s_{i_{\check{M}^1}^1}^1(t)\}$$

denote the skill vector for the entire team that is flying at time t, and similarly define skill vectors for sailing and swabbing teams with $\boldsymbol{\sigma}^2(t)$ and $\boldsymbol{\sigma}^3(t)$.

5. As regards measuring satisfaction, this is somewhat more complicated. In Wynn's original development, the satisfaction is a monotonic function of the skill. However, an agent assigned to a task in which he is uninterested may not demonstrate increased satisfaction. We therefore assume that each agent has a specified interest area, and that satisfaction is measured as a monotonic function of skill in that specific area. We denote the specified interest area of X_i as $a_i \in \{1, 2, 3\}$. For our purposes this will assumed to be fixed, but more generally might change over time. (For example, if an agent finds himself highly skilled in an area different from his specified area, he might determine greater satisfaction by switching.)

   Let $g_i(t)$ denote the satisfaction of X_i at time t. Then

$$g_i(t) = \begin{cases} \hat{g}_i(s_i^{a_i}(t)) & \text{if } i \in I^{a_i}(t) \\ \check{g}_i(s_i^j(t)) & \text{if } i \in I^j(t), j \neq a_i, \end{cases} \tag{4.1}$$

   where it is assumed that $\hat{g}_i(x) \geq \check{g}_i(x)$ (greater satisfaction for preferred skills).

   Let $\mathcal{I}^j$ denote "interest group" j, by

$$\mathcal{I}^j = \{i \in (1, \ldots, N) : a_i = j\}.$$

6. Risk: From the management's point of view, there may be risk involved in an open process. In particular, in an "agent's market," an agent who become fed up with the way of doing business may simply walk away from the table. This does not necessarily coincide with a military way of doing business, but does introduce an interesting way of measuring the cost of holding out. Specifically, the following might be elements of risk:

   - There is the risk that agents may lose interest if the negotiations proceed for too long.

- Risk may be tied to perceived discrepancies: an agent perceiving favoritism directed toward another agent may become disenchanted.

7. Each mission incurs some danger or risk to the participants. We model that the group is as vulnerable as its least-skilled member. Denote the risk for the group performing task j at time t by $r^j(\min_{i \in I^j(t)} s_i^j(t))$, where r^j is a nondecreasing function of its argument. This represents the fact that the more skilled the group, the less susceptible to hazard.

8. As a result of participating in a mission in area j at time t, X_i's skills increase in that area as a function of the success of the group and the participant's skill level at that time. If X_i does not participate at time t, its skill atrophies. We write

$$s_i^j(t+1) = \begin{cases} f_1^j[g(\boldsymbol{\sigma}^j(t), s_i^j(t)] & i \in I^j(t) \\ f_2^j[s_i^j(t)] & i \notin I^j(t). \end{cases}$$

9. The individual goal of each agent is to increase its skill level in its specified area, or equivalently, to increase its satisfaction in its specified area.

10. The group goal is twofold. First, all necessary tasks must be completed. (That is, the group identifies with administrative requirements.) Second, as a group all participants are to increase their skill levels, particularly in their preferred area.

11. Each agent has some input on whether they participate on a mission. This is weaker than Wynn's model, in which all agents must agree on who participates on each mission. (Need to develop this issue more fully.)

12. "The group goal is primarily for all participants to increase their skill levels uniformly, that is, for them to achieve some form of skill equilibrium. Although this goal is generally consistent with the individual goals, it is not necessarily served by having each of the agents pursue their individual goals separately. There must be some principle of coordination that will supersede the individual interests. Let

$$g[\boldsymbol{\sigma}(t)] = g(s_{i_1}(t), s_{i_2}(t), \ldots, s_{i_M}(t)], i_j \in I(t), j = 1, 2, \ldots, M$$

denote the joint satisfaction for the group." (Wynn)

In the present case, the group goal needs to be reconsidered. More correctly, it is not desirable to have all skill levels of all agents approach uniformity in all areas, since it is expected that better skill will be obtained by specialists in a skill area. From a management perspective, the goal is to have enough agents skilled in each area to be able to effectively meet mission requirements, and to attain a modicum of agent satisfaction. Let

$$g[g_i(t), i = 1, \ldots, N]$$

denote the group satisfaction. This will be affected by the individual satisfaction on the tasks which the agents are participating in (depending on their preferences).

Let us take the selectability measure of X_0 to be the degree to which the mandatory jobs are filled, and the rejectability to be a measure of the un-skill of the agents involved. Thus, X_0 wants the job done, and prefers that it be done only by skilled agents. Beyond this, management does not care explicitly about the comfort of the other agents. (However, there is a built-in linkage: since better skilled agents are preferred, this should ultimately favor the building of skills in the agents, and so works in accord with their selectability function.)

We will form

$$\tilde{\mathfrak{s}}_0(\mathbf{u}(t), t) = \sum_{j=1}^{3} \max(\overline{M}^j - \check{M}^j, 0)$$

This measure counts as a penalty a failure to fill the mandatory $\overline{M}^j$ jobs, but does not incur any penalty for having more than this filled. Take

$$\mathfrak{s}_0(\mathbf{u}, t) = \max_{\mathbf{u}(t) \in \mathcal{U}(t)} \tilde{\mathfrak{s}}_0(\mathbf{u}(t), t) - \tilde{\mathfrak{s}}_0(\mathbf{u}(t), t)$$

as X_0's unnormalized selectability, and form

$$p_{s_0}(\mathbf{u}) = \begin{cases} \frac{\mathfrak{s}_0(\mathbf{u},t)}{\sum_{\mathbf{v} \in \mathcal{U}} \mathfrak{s}_0(\mathbf{u},t)} & \mathbf{u} \in \mathcal{U} \\ 0 & \text{otherwise} \end{cases}$$

Let the combined risk be denoted as

$$\mathfrak{r}_0(\mathbf{u}) = \rho_1 r^1(\min_{i \in I^1(t)} s_i^1(t)) + \rho_2 r^2(\min_{i \in I^2(t)} s_i^2(t)) + \rho_3 r^3(\min_{i \in I^3(t)} s_i^3(t))$$

where ρ_i are weighting factors for the various tasks, with $\sum_i \rho_i = 1$. For example, management may care less about the risk to swabbers than they do about the risk to pilots. The rejectability function p_{R_0} is formed by normalizing $r(\mathbf{u})$. This can be converted into a rejectability (if desired) by normalization:

$$p_{R_0}(\mathbf{u}) = \begin{cases} \frac{\mathfrak{r}_0(\mathbf{u})}{\sum_{\mathbf{v} \in \mathcal{U}} \mathfrak{r}_0(\mathbf{v})} & \mathbf{u} \in \mathcal{U} \\ 0 & \text{otherwise.} \end{cases}$$

The individual selectability for an agent $X_i, i = 1, 2, \ldots, N$ is determined by

$$\mathfrak{s}_i(\mathbf{u}) = g_i(t)$$

where $g_i(t)$ is defined by (4.1). The individual rejectability for X_i is

$$\mathfrak{r}_i(\mathbf{y}) = r^j(s_i^j(t))$$

where $r^0(x) = 0$ (no risk in doing nothing).

4.4.2 Resource and job descriptions

At time t, assume that there are R^j resources available to accomplish the M^j tasks (one resource for each task). It may occur that $M^j > R^j$ (more tasks than there are resources).

In an eventual description, there may be a history associated with each resource, and some criterion for resource use. For example, it may be desirable to ensure that each plane has approximately the same amount of air time. However, we will not worry about such a complication at this point.

The skill vector associated with task j is

$$\boldsymbol{\sigma}^j(t) = \bigcup_{i:[\mathbf{u}(t)]_i=j} s_i^j(t).$$

Individual selectability: based on wanting to fly. Individual rejectability: tied to length of time negotiations take.

Joint selectability for $X_1, \ldots, X_N$: Each individual wants to fly, but also wants the group as a whole to succeed.

Joint selectability for X_0: see that the necessary tasks are filled. Keep the agents as happy as possible.

How to tie these together? Let $I = [0, 1, \ldots, N]$, $\overline{I} = [1, \ldots, N]$ (excluding the first manager index). Then

$$p_{s_I} = p_{s_0, s_{\overline{I}}} = p_{s_0|s_{\overline{I}}} p_{\overline{I}}$$

$p_{s_0|s_{\overline{I}}}$ represents the manager's selectability given the agent's selectabilities. If he is recalcitrant or unresponsive, then his perceptions do not change with the selectabilities of the other agents, $p_{s_0|s_{\overline{I}}} = p_{s_0}$. This makes an interesting model.

Agent joint: Let us assume an enlightened stance: each agent is willing to concede to any less skilled than he. Let the skill levels $p_{\overline{I}}$

4.4.3 Negotiation

(from Wynn) With the joint selectability and rejectability functions, we form the joint satisficing set

$$\mathbf{S}_b(t) = \{\mathbf{u} \in \mathcal{U}\colon p_{\mathbf{s}}(\mathbf{u}, \mathbf{s}(t)) \geq b p_{\mathbf{R}}(\mathbf{u}, \mathbf{s}(t))\}$$

Marginal selectability and rejectability:

$$p_{S_i}(u_i, \mathbf{s}(t)) = \sum_{u_j \in U_j} p_{S_1,\ldots,S_N}(u_1, \ldots, u_N, \mathbf{s}(t))$$

$$p_{R_i}(u_i, \mathbf{s}(t)) = \sum_{u_j \in U_j, j \neq i} p_{R_1,\ldots,R_N}(u_1, \ldots, u_N, \mathbf{s}(t))$$

Individually satisficing sets:

$$S_b^i(t) = \{u_i \in U_i\colon p_{S_i}(u_i; \mathbf{s}(t)) \geq b p R_i(u_i; \mathbf{s}(t))$$

Then the individually satisficing sets and the jointly satisficing sets can be reconciled using negotiation, as discussed above.

4.5 Satisficing negotiation for resource allocation with disputed resources

In this section, we present a resource allocation study in which resources are disputed. This material comes from [21].

Decision making agents acting together should be influenced not only by their own aspirations and budgets but by these aspects of other agents in the system. To represent this interaction among agents, a notion of *group* rationality must be embodied in the decision systems of interacting agents. Group rationality is not necessarily a logical consequence of rationality based on individual self interest. Under a model of rationality in which maximization of utility is the operative notion, group behavior obtained by amalgamation of the individual behaviors is not usually optimized by optimizing each individual behavior, as is typically done in a game-theoretic setting. Those who put their final confidence in the limited perspective of exclusive self-interest may ultimately function disjunctively, and perhaps illogically, when participating in collective activities. Rather than reorient game theory to accommodate situations where coordination is a more natural operational descriptor of the game than is self-interested conflict, we propose to describe notions that are neutral with respect to questions of conflict and coordination.

Beyond simply taking into account the presence of other acting agents in a system, there is frequently some form of sociality that is conducive to at least a weak form of congruity or mutual agreement. In cooperative scenarios, agents agree to work together; in competitive scenarios, agents tacitly agree to oppose each other. The procedures used to arrive at these agreements are not determined simply as a function of the preference structure of the decision makers, whether posed in a framework of self interest or community interest. Agreement among agents is typically obtained via a process of negotiation, in which multiple agents evaluate and share information when they have incentive to strike a mutually acceptable compromise. In the negotiating process, it is not sufficient for a decision maker merely to identify an acceptable joint solution (for the community) according to its own lights. The entire community should "buy into" a joint solution that is mutually acceptable.

In negotiation it is rare that all parties involved will tip their hands to reveal all of the factors influencing their decisions. In a competitive setting, a policy of secrecy may keep competitors from exploiting a weakness, or it may be used to persuade competitors to a more advantageous position. In a cooperative setting, complete disclosure of information might be precluded due to restrictions in communication bandwidth and/or time. Because of a lack of disclosure, negotiation may invoke principles of inference, wherein agents attempt to estimate positions or attitudes of other agents based on the options they bring to the bargaining table.

In light of these observations, some principles of negotiation are suggested:

N-1 Negotiators must typically be concerned with meeting minimum requirements more than achieving maximum performance.

N-2 Negotiations should lead to decisions that are both good enough for the group as a whole (as established by a group rationality) as well as good enough for each individual (as established by local preferences).

N-3 Negotiation is typically an iterative process. Starting from a set of initial joint options, it is natural to iterate toward solutions which are individually acceptable, rather than attempting to move directly to joint options which are a best compromise.

N-4 Negotiation may frequently incorporate elements of inference.

A rich model for negotiation should be able to capture other aspects of the negotiation process, such as recalcitrance (resistance to accede to group preferences), accommodation (openness to group preferences), or annoyance over extended or unchanging negotiation positions.

In this paper, we will briefly review the concept of praxeic utility decision theory as a means of implementing satisficing control, then extend this to multiple agent decision making to model group rationality. Concepts of negotiation consistent with the principles outlined above are established using this multi-agent satisficing framework. As case study of a problem for which a negotiated solution is reasonable, a problem of resource allocation with disputed resources is modeled.

4.5.1 Satisfi cing decision making: single and multiple agents

Single agent satisficing

Satisficing, a term coined by Simon [22], refers to a decision making strategy in which options are selected which are "good enough," differing thereby from conventional approaches which seek only the best. From the satisficing viewpoint, being "good enough" is sufficient; insisting on the best and only the best via an optimizing algorithm may be an overly restrictive luxury. From an operational point of view, however, while establishing that an option is (at least locally) optimal is at least expressible as a optimization problem, establishing what is "good enough" appears to be more elusive. The question of establishing good enough choices is addressed from a philosophical point of view with regard to truth systems by Levi [23, 24, 25]. In this epistemological framework, known as epistemic utility, options are sought for which the amount of information associated with them exceeds the potential for error. All options are deemed acceptable — good enough — that pass a likelihood ratio test comparing a truth valuation (a probability) and an informational value of rejection (also constructed as a probability). Application of epistemic utility to control problems yields *praxeic utility theory* [26, 27, 28, 29, 30, 31, 32, 33]. (For a discussion of praxeic utility theory in the context of negotiation, and for a more complete development of these concepts, see [34].) In this theory, a selectability function $p_S(u)$ is formed which, for each option u available from a universe of options U available to a decision making agent X, measures the degree to which u works toward success in achieving the agent's goals. Also, a rejectability function $p_R(u)$ is established which measures costs associated with each option. This pair of measures, called collectively the *satisfiability functions*, are endowed with the mathematical structure of probabilities (e.g., they are nonnegative and sum to 1 on the U).

Definition 4.1 The *satisficing set* Σ_q is the set of options defined by

$$\Sigma_q = \{u \in U \colon p_S(u) \geq q p_R(u)\}. \tag{4.2}$$

□

The satisficing set consists of those options for which the benefit exceeds the cost: the set of alternative which are arguably "good enough." There may be more than one option in Σ_q. Moving away from strict adherence to optimality increases the flexibility, while by not retaining only the best. Ultimate selection of a single option for action is accomplished by means of a tie breaking rule, such as most selectable, least rejectable, or maximally discriminating.

The parameter q in (4.2) is the *index of caution*. As q is increased, fewer options are accepted into the the satisficing set. As such, the agent exhibits greater caution, accepting only options of higher merit in comparison to their cost. We say that Σ_q is the satisficing set at level q. Because of its similarity to likelihood ratio tests in conventional decision theory, the test in (4.2) is referred to as the *praxeic likelihood ratio test* (PLRT).

Multiple agent satisficing

Satisficing decision theory extends very naturally to multiple agent systems. Satisficing admits degrees of fulfillment, whereas optimization is an absolute concept. While the statement "What is best for me and what is best for you is also jointly best for us together" may be nonsense, the statement "What is good enough for me and what is good enough for you is also jointly good enough for us together" may be perfectly sensible, especially when we do not have inflexible notions of what it means to be "good enough." Satisficing grants room for compromise, leaving open the opportunity for one or more agents involved to relax standards of individual performance in the interest of the good of the community. A theory of multi-agent satisficing thus provides the stage on which the act of negotiation can reasonably be presented.

Since they possess the mathematical structure of probabilities, selectability and rejectability can be naturally extended to the multivariate case by defining *joint* selectability and rejectability measures, which may be used to determine a jointly satisficing set. In addition, individual decision makers may establish individual notions of satisficing by computing marginal selectability and rejectability functions from the joint expressions.

Let $X_1, \ldots, X_N$ be N interacting agents, where each agent has its own decision space U_i. The joint action space is the space $\mathbf{U} = U_1 \times U_2 \times \cdots \times U_N$. A joint decision is an element $\mathbf{u} = (u_1, u_2, \ldots, u_N) \in \mathbf{U}$. We denote the ith element of $\mathbf{u}$ as $\mathbf{u}(i)$.

An act by any single member of a multi-agent system has potential ramifications throughout the entire community. And, although a participant may perform an act either in its own interest or for the benefit of others, the act is usually not implemented free of cost: resources are expended or risk is taken, perhaps by the single agent, but also perhaps by the entire community. Although these consequences may be defined independently from the benefits, the measures associated with benefits and cost cannot necessarily be specified independently of each other. In light of this, the object representing the relationships between agents in their systems regarding their individual and joint selectability and rejectability is an *interdependence measure* which combines both rejectability and selectability for all agents as a multivariate probability function of the form

$$p_{S_1,S_2,\ldots,S_N,R_1,R_2,\ldots,R_N}(u_1, u_2, \ldots, u_N, v_1, v_2, \ldots, v_N).$$

This is expressed more briefly as $p_{\mathbf{S},\mathbf{R}}(\mathbf{u},\mathbf{v})$. Values for the interdependence measure are typically obtained by means of factorization into constituent conditional and marginal probabilities. In these factorizations, agents may represent how their selectabilities or rejectabilities are affected by the selectability and rejectability of other agents. From the general interdependence function, the joint selectability function is obtained by a marginalization

$$p_{\mathbf{S}}(\mathbf{u}) = \sum_{\mathbf{v}\in\mathbf{U}} p_{\mathbf{S},\mathbf{R}}(\mathbf{u},\mathbf{v})$$

and similarly

$$p_{\mathbf{R}}(\mathbf{v}) = \sum_{\mathbf{u}\in\mathbf{U}} p_{\mathbf{S},\mathbf{R}}(\mathbf{u},\mathbf{v})$$

Definition 4.2 The *multipartite satisficing decision rule* defines the set *multipartite satisficing set* by

$$\boldsymbol{\Sigma}_q = \{\mathbf{u}\in\mathbf{U}\colon p_{\mathbf{S}}(\mathbf{u}) \geq q p_{\mathbf{R}}(\mathbf{u})\}. \tag{4.3}$$

□

Joint options in $\boldsymbol{\Sigma}_q$ are those for which the benefits exceed the costs, as viewed from the perspective of the group and as represented by the joint selectability and joint rejectability. The test in (4.3) is referred to as the *joint praxeic likelihood ratio test* (JPLRT).

Given joint selectability and joint rejectability, an individual agent can compute marginals by

$$p_{S_i}(u_i) = \sum_{\{\mathbf{u}\in U\colon\, \mathbf{u}(i)=u_i\}} p_{\mathbf{S}}(\mathbf{u})$$

$$p_{R_i}(v_i) = \sum_{\{\mathbf{v}\in U\colon\, \mathbf{v}(i)=v_i\}} p_{\mathbf{S}}(\mathbf{v}).$$

The resulting individually satisficing set for X_i is then

$$\Sigma_q^i = \{u_i \in U_i\colon p_{S_i}(u_i) \geq q p_{R_i}(u_i)\}.$$

Alternatively, an agent could employ a p_{S_i} and p_{R_i} not obtained as a marginal, presenting a different face to the public than what it holds for itself, and use these function to compute Σ_q^i.

We will use the notation $u = \mathbf{u}(i)$ to indicate that the option $u \in U_i$ is the ith element of a joint option vector $\mathbf{u}$.

An option u that is jointly satisficing for X_i, is not necessarily individually satisficing for X_i. That is, a joint option $\mathbf{u}\in\boldsymbol{\Sigma}_q$ does not necessarily have $\mathbf{u}(i)\in\Sigma_q$. The converse, however, is true: if $u\in\Sigma_q^i$, then $u = \mathbf{u}(i)$ for some $\mathbf{u}\in\boldsymbol{\Sigma}_q$. This is established by the following.

Theorem 1 *(The negotiation theorem) If u is individually satisficing for X_i, then it must be the ith element of some jointly satisficing vector $\mathbf{u}$, i.e., $u = \mathbf{u}(i)$ for some $\mathbf{u}\in\boldsymbol{\Sigma}_q$.*

Proof Without loss of generality, assume $i = 1$. Let $u \in \Sigma_q^1$ (i.e., $p_{S_1}(u) \geq qp_{R_1}(u)$). To establish proof by contradiction, assume that $u \neq \mathbf{u}(i)$ for all $\mathbf{u} \in \mathbf{\Sigma}_q$. It follows that for all $\mathbf{v} \in U_2 \times \cdots \times U_N$, $p_{\mathbf{S}}(u, \mathbf{v}) < qp_{\mathbf{R}}(u, \mathbf{v})$. Then

$$p_{S_1}(u) = \sum_{\mathbf{v}} p_{\mathbf{S}}(u, \mathbf{v}) < q \sum_{\mathbf{v}} p_{\mathbf{R}}(u, \mathbf{v}) = qp_{R_1}(u),$$

which contradicts $u \in \Sigma_q$. $\square$

On the basis of the negotiation theorem, it may be argued that each agent has a seat at the negotiation table. No one is necessarily frozen out of a deal.

It is important to emphasize what the negotiation theorem does not provide. If u_1 is individually satisficing for X_1, and u_2 is individually satisficing for X_2, then by the theorem $u_1 = \mathbf{u}(1)$ and $u_2 = \tilde{\mathbf{u}}(2)$ for some $\mathbf{u}, \tilde{\mathbf{u}} \in \mathbf{\Sigma}_q$, However, it is not necessarily the case that $\mathbf{u} = \tilde{\mathbf{u}}$: the options that are both individually and jointly satisficing may be different for different agents. Thus, the negotiation theorem does not establish a "solution" to the problem. However, it provides the basis upon which a solution may be sought through an iterative negotiation scheme. To obtain buy-in from all agents, the options that are individually satisficing for each agent must be elements of the same jointly satisficing options. Such options are the result of negotiation.

The proof of the negotiation theorem makes use of the fact that the same index of caution is used to compute Σ_q^i as is used for $\mathbf{\Sigma}_q$, but an agent could use a higher q to determine $\mathbf{\Sigma}_q$ than it does for its own satisficing set. The negotiation theorem is not necessarily true in this case. Or it may happen that each agent has its own perception of the interdependence function. In this case, we denote X_i's interdependence function as $p^i_{\mathbf{S},\mathbf{R}}$, and the corresponding joint selectability and rejectability as $p^i_{\mathbf{S}}$ and $p^i_{\mathbf{R}}$. Again, the negotiation theorem applies to each agent separately, but not collectively. Another possibility is that an agent may use joint functions $p^i_{\mathbf{S}}$ and $p^i_{\mathbf{R}}$ for establishing $\mathbf{\Sigma}^i_q$, but use individual p_{S_1} and p_{R_1} not computed as marginals of $p_{\mathbf{S}}$ and $p_{\mathbf{R}}$, thereby presenting a different "public face" and "private face." Again, the negotiation theorem does not apply. (This raises the the question for future investigation: to what degree can these private satisfiability functions differ from marginal satisfiability functions and still have reasonable negotiations.)

The negotiation theorem, and these observations about its provisional application, motivate the development of algorithms for negotiation based on satisficing.

4.5.2 Satisficing Negotiation

Multi-agent satisficing is suited to the principles of negotiation outlined in the introduction. By admitting degrees of fulfillment, satisficing agents can explore options which are both mutually and individually good enough. In a negotiation process, however, it is not sufficient for a decision maker to identify a solution, even one which it views as being jointly acceptable. As mentioned above, other agents in the system may have their own models of the joint interdependence function, and their own individual satisficing functions, or may determine individually and jointly satisficing options not coincident with those of other agents. A negotiated solution should ideally be one in which all members of the community can individually concur. The multipartite satisficing set $\mathbf{\Sigma}_q$ and the individually satisficing set Σ^i_q provide each X_i with a basis for negotiation: an assessment from each agent's point of view of all options that are good enough for the group, and of the assessment of all individual options that are good enough for itself.

Compromise among a group of agents involves a lowering of standards by admitting possible actions that an agent, acting only unilaterally, would not necessarily prefer, but which it is willing to admit in the interest of acting as part of a group. The lowering of standards motivates the use of a satisficing outlook in the decision theory. An approach based on optimization, particularly one formulated on the basis of exclusive self-interest, does not admit grades or degrees. A choice is either optimal, or it is not. Compromise does not necessarily entail complete abolition of any agent's standards. An agent feeling that too much compromise is imposed it may walk away from the negotiating table.

In the formalism of multi-agent satisficing, an agent's index of caution q_i acts as a parameter representing the degree of compromise an agent is willing to adopt. By lowering the degree of caution, an agent is willing to consider placing more options in its satisficing set. As $q_i \to 0$, every option available to X_i is satisficing for X_i. If all agents are willing to sufficiently reduce their standards, a jointly acceptable solution can be obtained.

We will let $\mathbf{q} = (q_1, \ldots, q_N)$ denote the caution vector of the players. The least cautious index is $q_L = \min\{q_1, \ldots, q_N\}$. From an individual perspective, the negotiation theorem applies if an agent uses its own index of caution, q_i to determine the individually satisficing set, but uses q_L to determine its jointly satisficing set. It is assumed hereafter that each agent uses $q = q_L$ to determine $\mathbf{\Sigma}_q$. This reflects the conservative observation that the standards of a group can be no higher than the standards of any member of the group.

The relationship between individually and jointly satisficing sets for an agent is formalized by the following:

Definition 4.3 The set of all jointly satisficing vectors in $\mathbf{\Sigma}_q^i$ that are also individually satisficing for X_i is the *compromise set* $\mathbf{C}_i$, defined by

$$\mathbf{C}_i = \{\mathbf{u} = \{u_1, u_2, \ldots, u_N\} \in \mathbf{\Sigma}_{q_L}^i \colon u_i \in \Sigma_{q_i}\}.$$

$\square$

Since $q_L \leq q_i$ the negotiation theorem indicates that $\mathbf{C}_i$ is not empty. By the negotiation theorem, if $u \in \Sigma_{q_i}$, then there is some $\mathbf{u} \in \mathbf{C}_i$ such that $u = \mathbf{u}(i)$.

We define

$$\mathbf{C}_i(j) = \{u_j \colon u_j \in \mathbf{u} \text{ for some } \mathbf{u} \in \mathbf{C}_i\}$$

as the set of all options for X_j in $\mathbf{C}_i$.

Definition 4.4 The *joint accord set* $\mathbf{N}_{q_L}$ is the set of all vectors that are jointly (at caution level q_L) and individually (at caution level q_i) satisficing for all agents. That is,

$$\mathbf{N}_{q_L} = \bigcap_{i=1}^{N} \mathbf{C}_i.$$

Any joint option $\mathbf{u} \in \mathbf{N}_{q_L}$ is a *joint accord option*. $\square$

The desired outcome of a negotiation algorithm is a joint accord set. From $\mathbf{N}_{q_L}$, a single element joint accord option may be selected according to some tie breaking rule, such as the rule maximizing joint benefit to cost ratio,

$$\mathbf{u}^* = \arg \max_{\mathbf{u} \in \mathbf{N}_{q_L}} \frac{p_{\mathbf{S}}(\mathbf{u})}{p_{\mathbf{R}}(\mathbf{u})}. \tag{4.4}$$

This option is called the *rational compromise*.

If $\mathbf{N}_{q_L} = \emptyset$, then there are no decisions which are jointly and individually acceptable to all agents in the system.

Definition 4.5 In the context of multi-agent satisficing theory, *negotiation* is the process of working toward a solution which is individually and jointly acceptable to each agent in the system. □

Working toward a negotiated solution requires at least one of the agents to lower its standards, then recompute their compromise sets. If no agent is willing to compromise further (by lowering its own standards), then an impasse is reached in the negotiation process. Until that point is reached, however, negotiations may proceed in good faith. The lowering of standards is represented in this context by a lowering of an agents index of caution. Algorithm 4.1 outlines a negotiation algorithm based on this observation, termed the Enlightened Liberals algorithm ("liberal" in the sense of being tolerant of views other than one's own; "enlightened" in the sharing of information).

Algorithm 4.1 The Enlightened Liberals Negotiation Algorithm

Step 1: Initialize: $q_i = {}_0q_i$ for $i = 1, \ldots, N$. $q_L = \min({}_0q_i)$.

Step 2: X_i forms $\mathbf{\Sigma}^i_{q_L}$ and $\Sigma^i_{q_i}$.

Step 3: X_i forms $\mathbf{C}_i = \{\mathbf{u} \in \mathbf{\Sigma}^i_{q_L} \colon u_i \in \Sigma^i_{b_i}\}$.

Step 4: Communicate $\mathbf{C}_i$ and q_i to other agents.

Step 5: Each agent forms $\mathbf{N}_{q_L} = \cap_{i=1}^N \mathbf{C}_i$.

Step 6: If $\mathbf{N}_{q_L} = \emptyset$, each agent determines how much to lower q_i, then communicates q_i with the other agents. Then repeat from step 2.

Step 7: If $\mathbf{N}_{q_L} \neq \emptyset$, form the rational compromise $\mathbf{u} = (u_1, u_2, \ldots, u_N)$ according to (4.4).

In this algorithm, all agents communicate their choices in the same step. There is no way to use the partial information provided by another agent's compromise set to modify an agent's decisions.

Inference in negotiation

It may be noted that Enlightened Liberals is in accord with the first three principles of negotiation outlined in the introduction. However, no inference is employed in the algorithm as stated, since all agents essentially pass information simultaneously. The inference problem faced by agent

X_i is to estimate $p^j_{\mathbf{S}_j}$, $p^j_{\mathbf{R}_j}$, p_{S_j}, and p_{R_j} — the praxeic system employed by X_j — given the offered solutions brought to the negotiating table in the form of $\mathbf{C}_j$. As an estimation problem all the tools of statistical estimation theory can be brought to bear, such as Bayesian estimates, maximum likelihood, minimum variance, maximum entropy, etc. (The method selected is problem dependent.) In the example presented below, a heuristic is illustrated which is similar to maximum likelihood.

Incorporation of the inference aspect of the negotiation is outlined in algorithm 4.2, which differs from Enlightened Liberals mostly in the sequence nature of the exchange of information.

Algorithm 4.2 The Inferring Liberals Negotiation Algorithm

Step 1: Initialize: $q_i = {}_0q_i$ for $i = 1, \ldots, N$. $q_L = \min({}_0q_i)$.

Step 2: X_i infers updates for $p^i_{S_j}, p^i_{R_j}, p^i_{\mathbf{S}}$ and $p^i_{\mathbf{R}}$, based on the compromise sets for $\{\mathbf{C}_j, j = 1, 2, \ldots, N, j \neq i\}$.

Step 3: X_i forms $\mathbf{\Sigma}^i_{q_L}$ and $\Sigma^i_{q_i}$

Step 4: X_i forms $\mathbf{C}_i$.

Step 5: X_i communicates $\mathbf{C}_i$ and q_i to all other agents.

Step 6: After all agents have transmitted their information, each agent forms $\mathbf{N}_{q_L} = \cap_{i=1}^N \mathbf{C}_i$.

Step 7: If $\mathbf{N}_{q_L} = \emptyset$, each agent determines how much to lower q_i, then communicates q_i with the other agents. Then repeat from step 2.

Step 8: If $\mathbf{N}_{q_L} \neq \emptyset$, form the rational compromise $\mathbf{u} = (u_1, u_2, \ldots, u_N)$ according to (4.4).

4.5.3 Example: Disputed Resource Allocation

> Complexity is no argument against a theoretical approach if the complexity arises not out of the theory but out of the material which any theory ought to handle. — Frank Palmer
> *Grammar* (1971)

We illustrate the negotiation framework outlined above — including interdependence factorization, establishing satisfiability functions and inference — by means of a resource allocation problem. Consider a situation in which N agents are to allocate M resources among themselves in such a way that all resources are allocated to at least one agent, but more than one agent may claim a resource. A resource with more than one claimant is a *disputed* resource. Disputed resources are of lower value than undisputed resources, both because utilization of a resource is attenuated by virtue of sharing, and because of an intrinsic societal valuation that would avoid dispute. The goal of each agent is to obtain as much of the resources as possible (or the resource allocation with the maximum valuation), while working toward having the fewest disputed resources as possible. Starting from some initial allocation of resource, each agent must also sustain a cost of acquisition for each additional resource that is required. While expressed as an abstract “resource allocation” problem, it may be helpful to envision the geographical division of a country among non-aligned factions. The apportionment of the land of Israel among Israeli and Palestinian claimants is a recent motivating example. Interestingly, data that might be adapted for a problem on a larger scale for Europe in the mid-twentieth century have been the subject of research in studies in cooperation and complexity (see, e.g., [35]).

We will consider specifically only the two agent case; extensions to more agents is straightforward in principle. We will denote the allocation decision vector of agent X_i by a vector $u^i \in \{0,1\}^M$ where $u^i_j = 1$ if resource j is selected by agent i. (A final option denoted as $u^i = \emptyset$ might also be used to indicate that X_i is terminating the negotiation process and walking away from the negotiating table.) The decision vector $\overline{u}^i$ is used to indicate the Boolean complement of the decision vector u^i. For decision vectors u^1 and u^2, we will denote by $d = u^1 \cap u^2$ the disputed resources claimed by both agents[2] We denote by $u^i \backslash d$ the resources claimed exclusively by X_i. As agents begin the process, they have some initial allocation ${}_0u^1$ and ${}_0u^2$, with disputed resources

$${}_0d = {}_0u^1 \cap {}_0u^2.$$

Formulation of selectability and rejectability

The joint interdependence function is $p_{\mathbf{S},\mathbf{R}}(\mathbf{u},\mathbf{v}) = p_{S_1,S_2,R_1,R_2}(u_1,u_2,v_1,v_2)$. When we want to represent explicitly that this is the interdependence function as perceived by agent X_i, this will be denoted as $p^i_{S_1,S_2,R_1,R_2}(u_1,u_2,v_1,v_2)$. If this is changing with negotiation iteration number η, we will indicate this with $p^i_{S_1,S_2,R_1,R_2}(u_1,u_2,v_1,v_2;\eta)$.

To formulate specific results, it is expedient to factor the joint interdependence function into conditional and marginal probability measures. Conditional probabilities, as observed by Pearl [36] permit local or specific responses to be characterized. Conditional behavior is behavior at the local level, with all dependencies specified. Such factorizations permit characterization of global behavior in terms of local relationships, which are frequently easier to specify. A variety of factorizations are possible, even for the simple case of two agents, and it is not necessary for all agents to invoke the same factorizations. However, in this example, both agents will factor the joint interdependence function the same way.

We will express the factorization from the point of view of X_1, then express the inference process from the point of view of X_2 (using information from X_1). As a shorthand we will represent the factorization of the probability functions in terms only of their variables, using, for example, $S_2|R_2$ as a representation for $p_{S_2|R_2}(u_2|v_2)$. A reasonable (but not unique) factorization, expressed from the point of view of X_1 is

$$(S_1,S_2,R_1,R_2) = \\ (S_1|S_2,R_1,R_2)(R_1|S_2,R_2)(S_2|R_2)(R_2).$$

[2] The notation $u_1 \cap u_2$ might be more exactly represented as $u_1 \wedge u_2$, where a “bitwise” AND operation is implied in each element. However, the $\cap$ notation seems to be more suggestive.

Under the assumption that rejectability and selectability are independent for a given agent, we obtain

$$(S_1, S_2, R_1, R_2) = (S_1|S_2, R_2)(R_1|S_2, R_2)(S_2)(R_2) \tag{4.5}$$

There is an attractive symmetry in the first two factors, being the selectability and rejectability (respectively), conditioned on those quantities for the other agent. The first two terms of this factorization represent X_1's selectability and rejectability, respectively, when X_2 places all of its selectability mass and rejectability mass as the conditioning arguments. The factorization in (4.5) is expressed more explicitly as

$$p^1_{S_1,S_2,R_1,R_2}(u_1, u_2, v_1, v_2) = \\ p^1_{S_1|S_2,R_2}(u_1|u_2, v_2)p^1_{R_1|S_2,R_2}(v_1|u_2, v_2)p^1_{S_2}(u_2)p^1_{R_2}(v_2). \tag{4.6}$$

In the sections below, we describe the parameters that affect each of the terms in this factorization.

Goals

Each agent wants to maximize the value of the resources it claims. There is a functional $g^i(u)$ adopted by X_i evaluating its intended allocation. This might be quite simple, as in[3]

$$g^i(u) = \sum_{j \in u} \varepsilon^i(j),$$

where $\varepsilon^i(j)$ measures the intrinsic value of resource j. This may include the size of the resource as well as other attributes. (In the case of land as a resource, it might measure attributes such as a harbor or an airport, mineral or agricultural assets, or historical or religious value.) In the case that the resources are distributed in space, the value may be determined using less localized measures. For example, there may be more value in having the resources as near to each other as possible, or in a contiguous block. Or there may be less value to a resource which is surrounded by resources claimed by the other agent. (In the case of land apportionment, an agent might prefer large pieces contiguously joined, with no islands of other agents' land in the middle.) All of these variations can be incorporated into $g^i(u)$.

Given that the agents' goals are prescribed by the desire to obtain more resources, we simply normalize the allocation value to form a probability mass function:

$$p^1_{S_1|S_2,R_2}(u_1|u_2, v_2) = p^1_{S_1}(u_1) \propto g^1(u_1).$$

That is, the the selectability is conditionally independent of X_2's options. In the joint selectability each agent thus acts independently:

$$p_{S_1,S_2}(u_1, u_2) = p_{S_1}(u_1)p_{S_2}(u_2).$$

In (4.6), the term $p^1_{S_2}(u_2)$ is X_1's model (or perception) of X_2's selectability. This is determined simply X_1's estimate of $g^2(u)$ (estimated according to X_1's knowledge of X_2).

[3]The set $\{j \in u\}$ appearing in this summation is a shorthand for $\{j\colon u_j = 1\}$

Costs

Several elements of the problem contribute to an agent's perception of the cost of the choice.

Reduce disputed resource

Each agent evidences the difficulty of sharing the resource by seeking to eliminate the disputed resources. This not only serves his purposes — since disputed resources may not be enjoyed at full value — but also makes a concession to the society of the agents, which would prefer undisputed allocations.

In general, there is a cost function associated with disputed resources, which for agent X_i is denoted as $\delta^i(u^1 \cap u^2)$. This could depend on a variety of societal or historical factors. In some disputed resources, there may be no cost associated with more than one claim on the resources, whereas for others there is considerable cost.

A simple model for the cost is simply to make the disputation cost function proportional to the value of the disputed resource to each agent,

$$\delta^i(u^1 \cap u^2) = \delta^i(d) \propto \sum_{i \in d} \varepsilon_i^1 + \varepsilon_i^2.$$

This cost can be placed in the context of the conditional probability $p^1_{R_1|S_2,R_2}(u_1|u_2, v_2)$ as follows.

- When $u_2 = \overline{v}_2$, then X_2 places all of its selectability and none of its rejectability on the vector u_2, so it is fully committed to the option u_2. Then it may be presumed that there will be a disputation of $d = v_1 \cap u_2$, and the cost becomes $\delta^1(v_2 \cap u_2)$.

- When $u_2 = v_2$, then X_2 places all of its selectability as well as all of its rejectability on u_2, and hence is conflicted. In this conflicted state, X_1 assumes half the cost of disputed territory, $\frac{1}{2}\delta^1(v_2 \cap u_2)$. (Other options are, of course, possible to deal with this conflicted state.)

- For those territories that X_2 has indicated that it doesn't want (no selectability and high rejectability), there is no cost for a disputed territory.

An overall rejectability function based on disputation can be formulated by normalization. We will call this rejectability function $p^1_{R_1|S_2,R_2;\delta}(u_1|u_2, v_2)$.

Cost of Acquisition

There is a cost associated with acquiring the resources beyond the initial allocation. (For example, in the case of land resources, simply making the decision to acquire the land does not make it so. It may be necessary to deploy troops to enforce the decision, or to move in colonists, etc.) The cost of acquisition will also depend on the interest that the competing agent has in the new territory. In the context of the conditional probability $p^1_{R_1|S_2,R_2}(u_1|u_2, v_2)$, the following observations can be made.

- In the case that $u_2 = \overline{v}_2$ (that is, X_2 puts all of its selectability and none of its rejectability on the vector u_2), then it may be presumed that there will be a dispute over $d = v_1 \cap u_2$. The cost of the disputed acquisition is denoted by

$$\check{\chi}^1(d; {}_0u^1, {}_0u^2),$$

while the cost of the undisputed acquisition is

$$\hat{\chi}^1(v^1 \backslash d; {}_0u^1, {}_0u^2)$$

The total cost is then the sum of these:

$$\chi^1(v^1; {}_0u^1, {}_0u^2) = \hat{\chi}^1(v^1 \backslash d; {}_0u^1, {}_0u^2) + \check{\chi}^1(d; {}_0u^1, {}_0u^2)$$

- When $u_2 = v_2$; that is, X_2 is conflicted, placing all of its selectability as well as all its rejectability on u_2, then X_1 might assume that X_2 will not be in disputation; then

$$\chi^1(v^1; {}_0u^1, {}_0u^2) = \hat{\chi}^1(v^1; {}_0u^1, {}_0u^2)$$

- For those options on which X_2 places none of its selectability and all of its rejectability on, we will assume that same result as when X_2 is conflicted. (More generally, we could have a reduced cost for acquisition.)

- In the more general case, X_2 may be conflicted in some areas but not in others. In this case, X_1 only counts as disputed those territories which intersect with its interests and for which X_2 is unconflicted.

Combining these costs together and suitably normalizing, the rejectability function $p^1_{R_1|S_2,R_2;\chi}(u_1|u_2, v_2)$ is obtained.

Cost of negotiation

An agent may attribute cost to the process of negotiation. If the negotiation must proceed through several iterations, an agent may become sufficiently annoyed at the process that its response is to walk away from the negotiating table. Several factors may be incorporated into the cost of negotiation, including the number of iterations (which we denote by η), or the apparent lack of progress (if the compromise sets coming from other agents appears to be unchanging). A cost based on the number of iterations can also represent determination of an agent to with respect to certain options: while the overall boldness is decreasing, the rejectability of some options can be correspondingly increased to partially offset the reduction. The cost of negotiation is represented by $\alpha^i(u_1, u_2, v_2; \eta)$; suitably normalized it becomes the rejectability function $p^1_{R_1|S_2,R_2;\alpha}(u_1|u_2, v_2; \eta)$

Overall conditional rejectability

The conditional rejectability function in (4.6) is expressed as a convex sum of the rejectability functions described above:

$$p^1_{R_1|S_2,R_2}(u_1|u_2,v_2) = \beta_1 p^1_{R_1|S_2,R_2;\delta}(u_1|u_2,v_2) + \beta_2 p^1_{R_1|S_2,R_2;\chi}(u_1|u_2,v_2) + \beta_3 p^1_{R_1|S_2,R_2;\alpha}(u_1|u_2,v_2;\eta) \quad (4.7)$$

where $\sum_i \beta_i = 1$.

The marginal $p^1_{R_2}(v_2)$ and joint rejectability

The quantity $p^1_{R_2}(v_2)$ in (4.6) is X_1's model of X_2's marginal (unconditional) rejectability. This is viewed (in this formulation) as separate parameter, not a derived quantity. A variety of factors influence the joint rejectability. Even if the factors could be computed exactly, the weighting factors in the combination may be unknown. The difficulty of estimating this reliably suggests the need to estimate this quantity, if possible, during the negotiating process. Inference of $p^i_{R_j}(v)$ is discussed below.

As the negotiating process begins, some initial condition is needed. One initial condition reflecting this uncertainty is to assume that $p^1_{R_2}(v_2)$ apportions equal rejectability to all options. Another approach is to allow $p_{R_2}(v_2)$ — as an unconditional measure — to reflect those aspects of the problem that are most independent of actions or goals of other agents. In this light, allowing $p^1_{R_2}(v_2)$ to be proportional to the cost of acquisition is reasonable,

$$p^1_{R_2}(v_2) \propto \chi^2(v_2; {}_0u^2) + \delta^2(v_2, {}_0u^1 \cap {}_0u^2).$$

It is straightforward to verify that the joint rejectability can be computed as

$$p_{R_1,R_2}(v_1,v_2) = \sum_{u_2 \in U_2} p^1_{R_1|S_2,R_2}(v_1|u_2,v_2) p^1_{S_2}(u_2) p^1_{R_2}(v_2).$$

Inference

We consider now the question of inference of the parameters of other agents during the course of negotiation, presenting a method which is reasonable in the context of the present problem. After its initial decision-making step, X_1 presents $\mathbf{C}_1$ and q_1 to X_2. Based on this compromise set and caution index, what can be inferred about X_1's satisfiability functions? Because X_2 will be doing its computations based on the factorization (4.6), it may be assumed that that X_2 has a model of $p_{S_1}(u)$, since this is based primarily on economic questions which are observable by all agents. As mentioned above, however, $p^j_{R_i}(u)$ is difficult to obtain without further information. This parameter influences the joint rejectability $p^2_{R_1,R_2}$, and hence the marginal $p^2_{R_2}$, so its estimation has an extended influence in the decision making process. (This section, for the sake of definiteness, is presented as if X_2 were making inference based on information from X_1.)

Given $p^2_{S_1}(u)$ and $p^2_{R_1}(u)$, consider the joint options in $\mathbf{C}_1$. If $p^2_{S_1}(u) \geq q_1 p^2_{R_1}(u)$ and $u = \mathbf{u}(i)$ for some $\mathbf{u} \in \mathbf{C}_1$, then the compromise set provides no information: it reflects decisions that would be made by X^2 using its estimates. Also, if $p^2_{S_1}(u) < q_1 p^2_{R_1}(u)$ and $u \notin \mathbf{C}_i(i)$ then no additional information is provided: X_2 did not expect the choice, and X_1 did not select it.

However, if $p^2_{S_1}(u) \geq q_1 p^2_{R_1}(u)$ and $u \notin \mathbf{C}_i(i)$ then X_1 has rejected option, both individually and jointly, which according to X_2's model it should have accepted. Furthermore, if $p^2_{S_1}(u) < q_1 p^2_{R_1}(u)$ but $u \in C_i(i)$, then X_1 has accepted options both individually and jointly which, according to X_2's model it should have rejected. Both of these circumstances evince that X_2's model $p^2_{R_1}(u)$ is inaccurate at u and stands updating. Our inference rule is to change the rejectability $p^2_{R_1}(u)$ in such a way that these inconsistencies are resolved, and in such a way that the change at each point is minimized while ensuring that the probability constraint is satisfied.

Define the sets

$$\overline{U} = \left\{ u \in U_1 \colon \begin{array}{c} p^2_{S_1}(u) < q_1 p^2_{R_1}(u) \text{ and } u \notin \mathbf{C}_i(i) \\ \text{or} \\ p^2_{S_1}(u) \geq q_1 p^2_{R_1}(u) \text{ and } u \in \mathbf{C}_i(i) \end{array} \right\}$$

$$\hat{U} = \{u \in U_1 \colon p^2_{S_1}(u) < q_1 p_{R_1}(u) \text{ and } u \in \mathbf{C}_i(i)\}$$

$$\check{U} = \{u \in U_1 \colon p^2_{S_1}(u) \geq q_1 p_{R_1}(u) \text{ and } u \notin \mathbf{C}_i(i)\}$$

Elements in $\overline{U}$ have rejectability consistent with $\mathbf{C}_1$. Elements in $\hat{U}$ have rejectability too high to be consistent with $\mathbf{C}_1$, while elements in $\check{U}$ have rejectability too low. For $u \in \hat{U}$ or $u \in \check{U}$, form updated rejectability functions by

$$p^2_{R_1,\text{new}}(u) = \alpha(u) p^2_{R_1}(u)$$

where

$$\alpha(u) = \begin{cases} \frac{p^2_{S_1}(u) - \epsilon}{q_1 p^2_{R_1}(u)} & u \in \hat{U} \\ \frac{p^2_{S_1}(u) + \epsilon}{q_1 p^2_{R_1}(u)} & u \in \check{U} \end{cases}$$

for some small positive ϵ. This introduces a net change in rejectability

$$\Delta = \sum_{u \in \hat{U} \cup \check{U}} p_{R_1}(u)(1 - \alpha(u))$$

which is to be distributed among the rejectabilities of the elements in U_1 with the smallest change possible without affecting the decision boundaries. Let

$$\overline{U}_{\geq} = \{u \in \overline{U} \colon p^2_{S_1}(u) \geq q_1 p^2_{R_1}(u)\}$$

and let

$$\overline{U}_{<} = \{u \in \overline{U} \colon p^2_{S_1}(u) < q_1 p^2_{R_1}(u)\}.$$

The notation $|U|$ denotes the number of elements in the set U. Then the redistribution is as follows: If $\Delta > 0$ (i.e., rejectability is added):

- Distribute Δ among $\overline{U} \cup \check{U}$ as equally as possible, such that in $\overline{U}_{<} \cup \check{U}$, $p^2_{R1,\text{new}}(u) < 1$ and in $\overline{U}_{\geq}$, $p^2_{S_1}(u) \geq q_1 p^2_{R1,\text{new}}(u)$.

If $\Delta < 0$ (i.e., rejectability is removed):

- Distribute Δ among $\overline{U} \cup \hat{U}$ as equally as possible, such that in $\overline{U}_{\geq} \cup \hat{U}$ $p^2_{R1,\text{new}}(u) > 0$ and in $\overline{U}_{<}$, $p^2_{S_1}(u) < q_1 p^2_{R1,\text{new}}(u)$.

This simple-minded inference does not fully exploit the information available. For example, if by $\mathbf{C}_1$ X_1 appears uninterested in some resource, X_2 could parameterize an increase its interest in that area either by lowering a rejectability with respect to its acquisition, or by increasing its selectability. However, the inference described above suffices to demonstrate the concept.

4.5.4 Numerical demonstration

Consider a country with four regions as shown in figure 4.3. Values are apportioned in such a way that adjacent regions have a value greater than the sum of the constituent areas, and that value increases with more regions, as shown in table 4.1. The cost of disputed regions is also shown in table 4.1 (unnormalized). Cost of acquisition is on a region-by-region basis, as shown in table 4.2. The initial allocation is ${}_0u^1 = [1110]$ (countries 2, 3, and 4) and ${}_0u^2 = [1101]$. Figure 4.4 illustrates the sequence of estimated probabilities $p^1_{R_2}(u)$ and $p^2_{R_1}(u)$ for six iterations. (The abscissa represents the choice u as a decimal representation of the binary decision vector. The lines spaced within an integer $[u, u + 1)$ represent the probability estimates for different iterations of the negotiation algorithm.) After several iterations of negotiation, the compromise sets shown in table 4.3 are obtained, and the joint accord set $\mathbf{N}$ join in table 4.4 is obtained, where the rational compromise is indicated with *. (The integers represent the decimal form of the corresponding binary option vectors). The individually satisficing sets are $\Sigma_{q_1} = \{14\}$ and $\Sigma_{q_2} = \{5, 9, 12, 13\}$. The final boldness reached is $\mathbf{q} = (1.3, 1.3)$, after starting at $\mathbf{q}_0 = (1.9, 1.9)$ and decrementing each time by $\Delta q = 0.05$. It is interesting to note that even after negotiation, for the given value/cost data, the two agents end up with disputed regions, and that the initial conditions still remain in the joint accord set. However, having gone through the negotiating process, while regions must be shared, the agents may feel that they have "bought in" to this circumstance, since these options are individually satisficing.

4.5.5 Discussion

Within this framework for negotiation there are several observations that may be made. In some human negotiations, parties often repeat a position repetitively, without an apparent change of state, until at some point there is an abrupt change in feasible options. The procedure represented here provides a model for such behavior: even when from one iteration to the next there might be no change in compromise sets, each agent is modifying its models of the other agent, lowering its caution, and potentially changing its rejectability as a function of the number of iterations.

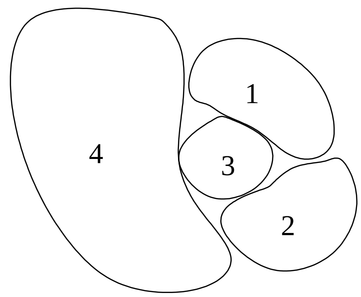


Figure 4.3: Four resources to be distributed

u	$\mathbf{u}$	$p^1_{S_1}(u)$	$p^2_{S_2}(u)$	$\delta^1(u)$	$\delta^2(u)$
0	0000	0	0	0	0
1	0001	0.0210	0.0227	3	3
2	0010	0.0252	0.0182	3	3
3	0011	0.0546	0.0500	7	7
4	0100	0.0420	0.0318	6	5
5	0101	0.0714	0.0636	8	9
6	0110	0.0756	0.0500	8	9
7	0111	0.1092	0.0818	15	12
8	1000	0.0252	0.0455	2	6
9	1001	0.0504	0.0727	4	9
10	1010	0.0546	0.0636	4	9
11	1011	0.0588	0.0727	9	5
12	1100	0.0714	0.0818	9	12
13	1101	0.1008	0.1045	13	14
14	1110	0.1050	0.0955	13	14
15	1111	0.1345	0.1455	20	20

Table 4.1: Valuations for resource allocation

u	$\check{\chi}^1$ (disp.)	$\hat{\chi}^1$ (undisp.)	$\check{\chi}^1$ (disp.)	$\hat{\chi}^1$ (undisp.)
0001	6	3	6	3
0010	4	2	4	2
0100	4	2	6	3
1000	8	4	10	5

Table 4.2: Cost of acquisition per country

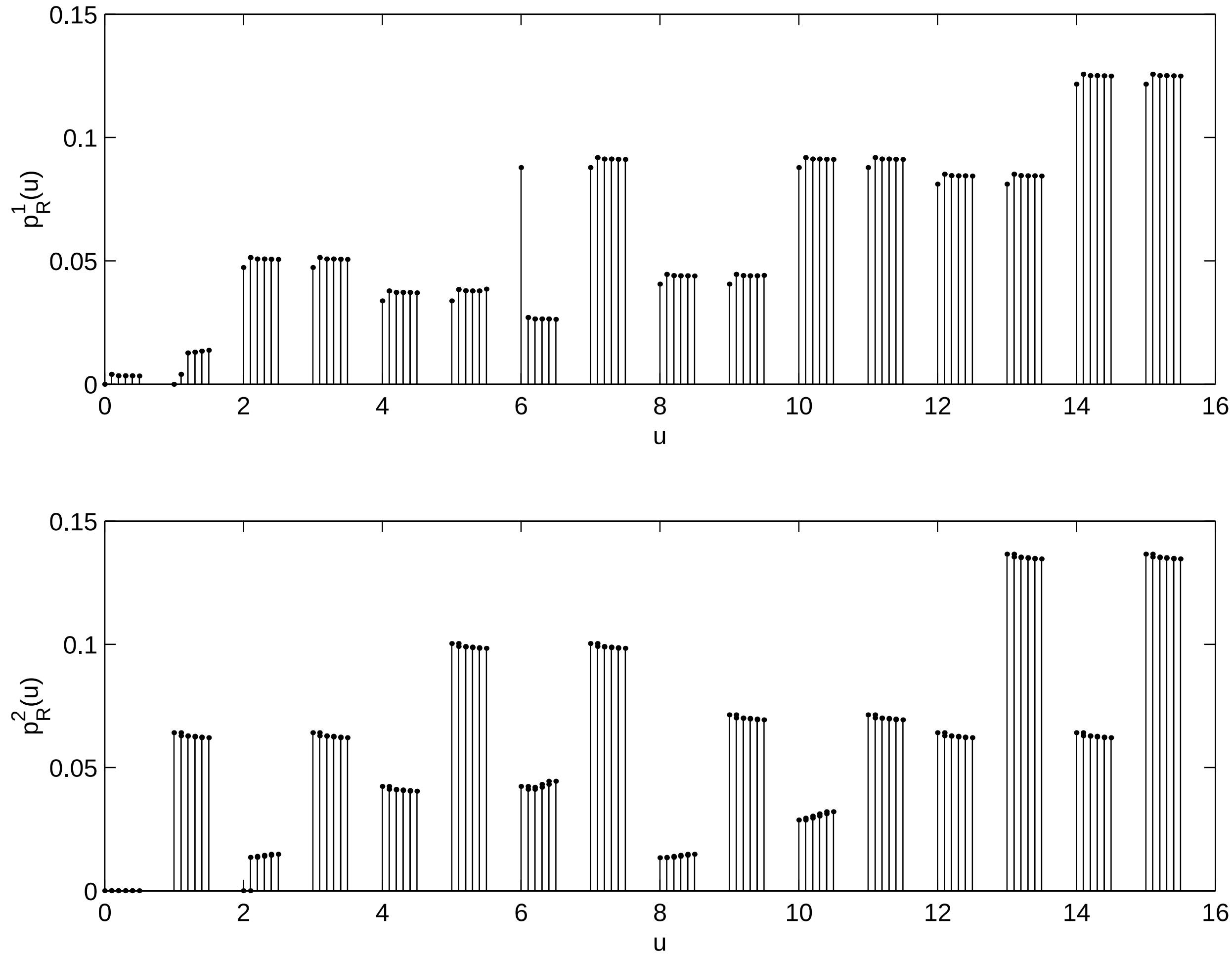


Figure 4.4: Sequence of estimated probabilities

$\mathbf{C}_1$	$\mathbf{C}_2$			
(14,1)	(2,5)	(4,9)	(15,9)	(2,13)
(14,3)	(4,5)	(6,9)	(2,12)	(4,13)
(14,4)	(5,5)	(7,9)	(4,12)	(6,13)
(14,5)	(6,5)	(8,9)	(6,12)	(7,13)
(14,6)	(7,5)	(9,9)	(7,12)	(8,13)
(14,8)	(8,5)	(10,9)	(8,12)	(10,13)
(14,9)	(12,5)	(11,9)	(10,12)	(12,13)
(14,12)	(14,5)	(12,9)	(12,12)	(13,13)
(14,13)	(2,9)	(13,9)	(14,12)	(14,14)
(14,15)	(3,9)	(14,9)	(15,12)	(15,13)

Table 4.3: Compromise sets after negotiation

N
$(14,5)^*$
(14,9)
(14,12)
(14,13)

Table 4.4: Joint accord set

Other behaviors such as recalcitrance or openness can be modeled depending on how the boldness is changed.

A concern that may be raised regarding this procedure is its computational complexity. A large measure of the complexity arises due to the computation of marginals in the formulation of p_{R_1,R_2}. The complexity can be mitigated somewhat by efficient organization of the computations, using, for example, the factor graph approach described in [37]. Another approach is to absorb the normalization used determining in p_{S_1,S_2} and p_{R_1,R_2} into the index of caution q. Once an initial q can be determined which provides for meaningful individual and joint solutions, the index of caution is adjusted until a group accord is established.

In conclusion, the multi-agent satisficing theory provides a means of describing solutions which are individually and jointly satisficing from the perspective of an individual agent in the community of agents. We have provided a definition of negotiation, which is the process of working to achieve accord among the different agents with regard to the solutions they they find acceptable, and provided some algorithms to implement that process. To demonstrate how the theory may be applied to a multi-agent problem, a resource allocation problem was presented in which agents vie for disputed resources.

4.6 A Praxeology for Rational Negotation

This section is drawn from [34].

4.6.1 Introduction

Negotiation is a branch of multi-agent decision making that involves the opportunity for repeated interaction between independent entities as they attempt to reach a joint decision that is acceptable to all participants. But unless the interests of the decision makers are extremely compatible, achieving such a compromise will usually require them to be willing to consider lowering their standards of what is acceptable if they are to avert an impasse. For an agent to consider lowering its standards, it must be willing to relax the demand for the best possible outcome for itself, and instead be willing to settle for an outcome that is merely good enough, in deference to the interests of others. Defining what it means to be good enough, however, is much more subtle than defining what it means to be optimal, and any such definition must be firmly couched in and consistent with the decision maker's concept of rationality.

Rational Choice

Fundamental rationality requires a decision maker to choose between alternatives in a way that is consistent with its preferences. Consequently, before a rational decision is possible, a decision maker must have some way to order its preferences.

Definition 4.6 Let the symbols "$\succeq$" and "$\cong$" denote binary relationships meaning "is at least as good as" and "is equivalent to," respectively, between members of a set $\mathcal{X} = \{x, y, z, \ldots\}$. The set $\mathcal{X}$ is *totally ordered* if relationships between elements of $\mathcal{X}$ are *reflexive* ($x \succeq x$), *antisymmetric* ($x \succeq y \;\; \& \;\; y \succeq x \implies x \cong y$), *transitive* ($x \succeq y \,\&\, y \succeq z \implies x \succeq z$), and *linear* (either $x \succeq y$ or $y \succeq x \;\forall\, x, y \in \mathcal{X}$). If the linearity condition is relaxed, then the set is *partially ordered*. □

Once in possession of a preference ordering, a rational decision maker must employ general principles that govern the way the orderings are to be used to formulate decision rules. Perhaps the most well-known principle is the classical economics hypothesis of [38] and [39], which asserts that individual interests are fundamental; i.e., that social welfare is an aggregation of individual welfares. This hypothesis leads to the doctrine of *rational choice*, the favorite paradigm of conventional decision theory. Rational choice is based upon two premises.

P-1 *Total ordering:* a decision maker is in possession of a total preference ordering for all of its possible choices under all conditions (in multi-agent settings, this includes knowledge of the total orderings of all other participants).

P-2 *The principle of individual rationality*: a decision maker should make the best possible decision for itself, that is, it should optimize with respect to its own total ordering (in multi-agent settings, this ordering will be influenced by the preferences of others).

A praxeology, or science of efficient action, is the philosophical underpinning that governs the actions of a decision-making entity. Conventional praxeologies are founded on the paradigm of rational choice. For single-agent systems, this equates to optimization, which typically results in maximizing expected utility. For multi-agent systems, rational choice equates to equilibration: a joint decision is an equilibrium if, were any individual to change its decision unilaterally, it would decrease its own expected utility. Rational choice has a strong normative appeal; it tell us what exclusively self-interested decision makers should do, and is the praxeological basis for much of current artificial decision system synthesis methodology. The ratiocination for this approach, as expressed by Sandholm, is that each decision maker should

> maximize its own good without concern for the global good. Such self-interest naturally prevails in negotiations among independent businesses or individuals . . . Therefore, the protocols must be designed using a *noncooperative*, *strategic* perspective: the main question is what social outcomes follow given a protocol which *guarantees that each agent's desired local strategy is best for that agent—and thus the agent will use it* [40, pp. 201,202].

This rationale is consistent with the conventional game-theoretic notion that society should *not* be viewed as a generalized agent, or superplayer, who is capable of making choices on the basis of some sort of group-level welfare function. So doing, [41] argues, creates an "anthropomorphic trap" of failing to distinguish between group choices and group preferences.

Anthropomorphisms aside, it is far from obvious that exclusive self interest is the appropriate characterization of agent systems when coordinated behavior is desirable. Granted, it is possible under the individual rationality regime for a decision maker to suppress its own egoistic preferences in deference to others by redefining its utilities, but doing so is little more than a device to trick individual rationality into providing a response that can be interpreted as unselfish. Such an artifice provides only an indirect way to simulate socially useful attributes of cooperation, unselfishness, and altruism under a regime that is more naturally attuned to competition, exploitation, and avarice. Luce and Raiffa summarized the situation succinctly when they observed that

> general game theory seems to be in part a sociological theory which does not include any sociological assumptions . . . it may be too much to ask that any sociology be derived from the single assumption of individual rationality [42, p. 196].

Often, the most articulate advocates of a theory are also its most insightful critics. Perhaps the essence of this criticism is that rational choice does not provide for the ecological balance that a society must achieve if it is to accommodate the variety of relationships that may exist between agents and their environment. But achieving such a balance should not require fabrication of a superplayer to aggregate individual welfare into group welfare. What it may require, however, is reconsideration of the claim that rational choice provides the appropriate praxeology for synthesizing cooperative social systems.

State of the Art

There are many proposals for artificial negotiatory systems under the rational choice paradigm, bounded in various ways to account for limitations in knowledge, computational ability, and negotiation time. [43] and [44] propose models of alternating offers; these approaches are refined by [45] to account for time constraints, and are further developed by [46, 47, 48], [49], and [50] to incorporate a time discount rate and to account for incomplete information via the introduction of a revelation mechanism. These approaches are based on a notion of *perfect equilibrium*, which is stronger than Nash equilibrium in that it requires that an equilibrium must be induced at any stage of the negotiation process. Similar manifestations of bounded rationality occur with [51], who present a general framework for metareasoning via decision theory to define the utility of computation. Others have followed these same lines (see, for example, [52], [53], and [54]), and yield optimal solutions according to performance criteria that is modified to account for resource limitations. Additional approaches to bounded rationality occur with [55], who provide a rational analysis framework that accounts for environmental constraints regarding what is optimal behavior in a particular context. Another individual rationality-based approach is to involve market price mechanisms, as is done by [56, 57], resulting in a competition between agents in a context of information service provision. [58] use the Clarke Tax voting procedure to obtain the highest sum of utilities in an environment of truthful voting. [59] present a method of "principled negotiation" involving proposed changes to an original master plan as a means of finding a distributed optimal negotiated solution.

Another stream of research for the design of negotiatory systems is to rely more heavily upon heuristics than upon formal optimization procedures. The approach taken by Rosenschein and Zlotkin is to emphasize special compromise protocols involving pre-computed solutions to specific problems [60, 61, 62, 63]. Formal models which describe the mental states of agents based upon representations of their beliefs, desires, intentions, and goals can be used for communicative agents [64, 65, 66, 67, 68, 69, 70]. In particular, Sycara develops a negotiation model that accounts for human cognitive characteristics, and models negotiation as an iterative process involving case-based learning and multi-attribute utilities [71, 72]. [73] provide logical argumentation models as an iterative process involving exchanges among agents to persuade each other and bring about a change of intentions. [74, 75] develop a negotiation framework that employs a Bayesian belief update learning process through which the agents update their beliefs about their opponent. [76] advance a notion of partial global planning for distributed problem solving in an environment of uncertainty regarding knowledge and abilities.

The above approaches offer realistic ways to deal with the exigencies under which decisions must be made in the real world. They represent important advances in the theory of decision making, and their importance will increase as the scope of negotiatory decision making grows. They all appear, however to have a common theme, which is, that if a decision maker could maximize its own private utility subject to the constraints imposed by other agents, it should do so. Exclusive self-interest is a very simple concept. It is also a very limiting concept, since it justifies ignoring the preferences of others when ordering one's own preferences. The advantage of invoking exclusive self-interest is that it may drastically reduce the complexity of a model of the society. The price for doing so is the risk of compromising group interests when individual preferences dominate, or of distorting the real motives of the individuals when group interests dominate. The root of the problem, in both of these extreme cases, is the lack of a way to account for both group and individual interests in a seamless, consistent way.

Middle Ground

Rather than searching for or approximating a narrowly defined theoretical ideal, an alternative is to focus on an approach that, even though it may not aspire to such an ideal, is ecologically tuned to the environment in which the agents must function. If it is to function in a coordinative environment, it should not ignore the possibility of distinct group interests, yet it must respect individual interests. It should be flexible with respect to evaluations of what is acceptable, yet it must not abandon all qualitative measures of performance. Kreps seems to be seeking such an alternative when he observes that

> ...the real accomplishment will come in finding an interesting middle ground between hyperrational behaviour and too much dependence on *ad hoc* notions of similarity and strategic expectations. When and if such a middle ground is found, then we may have useful theories for dealing with situations in which the rules are somewhat ambiguous [77, p. 184].

Is there really some middle ground, or is the lacuna between strict rational choice and pure heuristics bridgeable only by forming hybrids of these extremes? If non-illusory middle ground does exist, few have staked claims to that turf. Literature involving rational choice (bounded or unbounded) is overwhelmingly vast, reflecting many decades of serious study. Likewise, heuristics, rule-based decision systems, and various *ad hoc* techniques are well-represented in the literature. Rationality paradigms that depart from these extremes or blends thereof, however, are not in substantial evidence. One who has made this attempt, however, is Slote [78], who argues that it is not even necessary to define a notion of optimality in order to define a common sense notion of adequacy. He suggests that it is rational to choose something that is merely adequate rather than something that is best, and that moderation in the short run may actually be instrumentally optimal in the long run. Unfortunately, Slote does not metrize the notion of being adequate. It is far easier to quantify the the notion of bestness than it is to quantify the notion of adequacy. Striving for the best may be the most obvious way to use ordering information, but it is not the only way. This paper presents a notion of adequacy that is not an approximation to bestness—it is a distinct concept that admits a precise mathematical definition in terms of utility-like quantities. The motivation for pursuing this development is to soften the strict egoism of individual rationality and open the way for consideration of a more socially compatible view of rationality that does not rely upon optimization, heuristics, or hybrids of these extremes.

4.6.2 A New Praxeology

The assumption that a decision-maker possesses a total preference ordering that accounts for all possible combinations of choices for all agents under all conditions is a very strong condition, particularly when the number of possible outcomes is large. In multi-agent decision scenarios, individuals may not be able to comprehend, or even to care about, a full understanding of their environment. They may be concerned mostly about issues that are closest to them, either temporally, spatially, or functionally. A praxeology relevant to this situation must be able to accommodate preference orderings that may be limited to proper subsets of the community or to proper subsets of conditions that may obtain.

In societies that value cooperation, it is unlikely that the preferences of a given individual will be formed independently of the preferences of others. Knowledge about one agent's preferences may alter another agent's preferences. Such preferences are *conditioned on the preferences of others*. Individual rationality does not accommodate such conditioning. The only type of conditioning supported by individual rationality is for each agent to express its preferences conditioned on the choices of the others but not on their *preferences* about their choices. Each agent then computes its own expected utility as a function of the possible options of all agents, juxtaposes these expected utilities into a payoff array, and searches for an equilibrium. Although the equilibrium itself is governed by the utilities of all agents, the individual expected utilities that define the equilibrium do not consider the preferences of others. A praxeology for a complex society, however, should accommodate notions of cooperation, unselfishness, and even altruism. One way to do this is to permit the preferences (not just the choices) of decision makers to influence each other.

Tradeoffs

At present, there does not appear to be a body of theory that supports the systematic synthesis of multi-agent decision systems that does not rely upon the individual rationality premise. It is a platitude that decision makers should make the best choices possible, but we cannot rationally choose an option, even if we do not know of anything better, unless we know that it is good enough. Being good enough is the fundamental obligation of rational decision makers—being best is a bonus.

Perhaps the earliest notion of being "good enough" is Simon's concept of *satisficing*. His approach is to blend rational choice with heuristics by specifying aspiration levels of how good a solution might reasonably be achieved, and halting search for the optimum when the aspirations are met [22, 79, 80]. But it is difficult to establish good and practically attainable aspiration levels without first exploring the limits of what is possible, that is, without first identifying optimal solutions—the very procedure this notion of satisficing is designed to circumvent. Aspiration levels at least superficially establish minimum requirements, and specifying them for simple single-agent problems may be noncontroversial. But with multi-agent systems, interdependencies between decision makers can become complex, and aspiration levels can be conditional (what is satisfactory for me may depend upon what is satisfactory for you). The current state of affairs regarding aspiration levels does not address the problem of specifying them in multi-agent contexts. It may be that what is really needed is a notion of satisficing that does not depend upon arbitrary aspiration levels or stopping rules.

Let us replace the premise of individual rationality with a concept of being good enough that is distinct from being approximately best. Mathematically formalizing a concept of being good enough, however, is not as straightforward as optimizing or equilibrating. Being best is an absolute concept—it does not come in degrees. Being good enough, however, is not an absolute, and does come in degrees. Consequently, we must not demand a unique good-enough solution, but instead be willing to accept varying degrees of adequacy.

This paper proposes a notion for being good enough that is actually more primitive and yet more complicated to quantify than doing the best thing possible. It is a benefit-cost tradeoff paradigm of getting at least what one pays for. The reason it is more complicated to quantify is that it requires the application of two distinct metrics to be compared, whereas doing the best thing requires only one metric to be maximized. As a formalized means of decision making, this approach has appeared in at least two very different contexts: economics and epistemology—the former is intensely practical and concrete, the latter is intensely theoretical and abstract. Economists implemented the formal practice of benefit-cost analysis to evaluate the wisdom of implementing flood control policies [81]. The usual procedure is to express all benefits and costs in monetary units and to sanction a proposition if the benefits are in excess of the estimated costs. The problem with this concept, however, is that the individual interests are aggregated into a single monolithic interest by comparing the total benefits with the total costs. Despite its flaws, benefit-cost analysis has proven to be a useful way to reduce a complex problem to a simpler, more manageable one. One of its chief virtues is its fundamental simplicity.

A more sophisticated notion of benefit-cost appears in philosophy. Building upon the American tradition of pragmatism fostered by Peirce, James, and Dewey, [24] has developed a distinctive school of thought regarding the evolution of knowledge corpora. Unlike the conventional doctrine of expanding a knowledge corpus by adding information that has been justified as true, Levi proposes the more modest goal of avoiding error. This theory has been detailed elsewhere (see [24, 82, 33, 31, 26]). The gist is that, given the task of determining which, if any, of a set of propositions should be retained in an agent's knowledge corpus, the agent should evaluate each proposition on the basis of two distinct criteria—first, the credal, or subjective, probability of it being true, and second, the informational value[4] of rejecting it, that is, the degree to which discarding the option focuses attention on the kind of information that is demanded by the question. Thus, for an option to be admissible, it must be both believable and informative—all implausible or uninformative option should be rejected. Levi constructs an *expected epistemic utility function* and shows that it is the difference between credal probability and a constant (the index of caution) times another probability function, termed the informational-value-of-rejection probability. The set of options that maximizes this difference is the admissible set.

Single-Agent Satisficing

Levi's epistemology is to employ two separate and distinct orderings—one to characterize belief, the other to characterize value. This approach, originally developed for epistemological decision-making (committing to beliefs), may easily be adapted to the praxeological domain (taking action) by formulating praxeological analogs to the epistemological notions of truth and informational value. A natural analog for *truth* is *success*, in the sense of achieving the fundamental goals of taking action. To formulate an analog for informational value, observe that, just as the management of a finite amount of relevant information is important when inquiring after truth in the epistemological context, taking effective action requires the management of finite resources, such as conserving wealth, materials, energy, safety, or other assets. An apt praxeological analog to the informational value of rejection is the conservational value of rejection. Thus, the context of the decision problem changes from the epistemological issue of acquiring information while avoiding error to the praxeological issue of conserving resources while avoiding failure. To emphasize the context shift, the resulting utility function will be termed *praxeic utility*.

Let us refer to the degree of resource consumption as *rejectability* and require the rejectability function to conform to the axioms of probability. This new terminology emphasizes the semantic distinction of using the mathematics of probability in a non-conventional way. Thus, for a finite action space U, rejectability is expressed in terms of a mass function p_R: $U \to [0,1]$, such that $p_R(u) \geq 0$ for all $u \in U$ and $\sum_{u \in U} p_R(u) = 1$. Inefficient options (those with high resource consumption) should be highly rejectable; that is, if considerations of success are ignored, one should be prone to reject options that result in large costs, high energy consumption, exposure to hazard, etc. Normalizing p_R to be a mass function, termed the *rejectability mass function,* insures that the agent will have a unit of resource consumption to apportion among the elements of U. The function p_R is the dis-utility of consuming resources; that is, if $u \in U$ is rejected, then the agent

[4] Informational value, as used here, is distinct from the notion of "value of information" of conventional decision theory, which deals with the change in expected utility if uncertainty is reduced or eliminated from a decision problem.

conserves $p_R(u)$ worth of its unit of resources.

The degree that u contributes toward the avoidance of failure is the *selectability* of u. Let us define the *selectability mass function*, $p_S\colon U \to [0,1]$ as the normalized amount of success support associated with each $u \in U$. Suppose that implementing $u \in U$ would avoid failure. For any $A \subset U$, the utility of not rejecting A in the interest of avoiding failure is the indicator function $I_A(u) = \begin{cases} 1 & \text{if } u \in A \\ 0 & \text{otherwise} \end{cases}$. The *praxeic utility* of not rejecting A when u avoids failure is the convex combination of the utility of avoiding failure and the utility of conserving resources:

$$\phi(A,u) = \alpha I_A(u) + (1-\alpha)\left(1 - \sum_{v\in A} p_R(v)\right),$$

where $\alpha \in [0,1]$ is chosen to reflect the agent's personal weighting of these two desiderata—setting $\alpha = \frac{1}{2}$ associates equal concern for avoiding failure and conserving resources.

Generally, the decision-maker will not know precisely which u will avoid failure, and so must weight the utility for each u by the corresponding selectability, and sum over U to compute the expected praxeic utility.

$$\begin{aligned}\bar{\phi}(A) =& \sum_{u\in U}\left[\alpha I_A(u) + (1-\alpha)\left(1 - \sum_{v\in A} p_R(v)\right)\right] p_S(u) \\ =& \alpha \sum_{v\in A} p_S(v) - (1-\alpha)\sum_{v\in A} p_R(v) + (1-\alpha).\end{aligned}$$

Dividing by α and ignoring the constant term yields a more convenient but equivalent form:

$$\bar{\varphi}(A) = \sum_{u\in A}\left[p_S(u) - q p_R(u)\right],$$

where $q = \frac{1-\alpha}{\alpha}$. The term q is the *index of caution*, and parameterizes the degree to which the decision maker is willing to accommodate increased costs to achieve success. An equivalent way of viewing this parameter is as an index of boldness, characterizing the degree to which the decision maker is willing risk rejecting successful options in the interest of conserving resources. Nominally, $q = 1$, which attributes equal weight to success and resource conservation interests.

Definition 4.7 A decision maker is *satisficingly rational* if it chooses an option for which the selectability is greater than or equal to the index of caution times rejectability. □

We adopt this notion of satisficing as the mathematical definition of being *good enough*. The largest set of satisficing options is the *satisficing set*:

$$\Sigma_q = \arg\max_{A\subset U} \bar{\varphi}(A) = \{u \in U\colon p_S(u) \geq q p_R(u)\}. \tag{4.8}$$

Notice that (4.8) is in the form of a likelihood ratio test, since the selectability and rejectability functions are mass functions. Equation (4.8) is the *praxeic likelihood ratio test* (PLRT).

This concept of satisficing does not require that the set of good-enough solution be non-empty. If it is non-empty, however, fundamental consistency requires that the best solution, if it exists (under the same criteria), must be a member of that set.

Theorem 2 (a) $q \leq 1 \implies \Sigma_q \neq \emptyset$. (b) *If* $\Sigma_q \neq \emptyset$ *then there exists an optimality criterion that is consistent with* p_S *and* p_R *such that the optimal choice is an element of* Σ_q.

Proof (a) If $\Sigma_q = \emptyset$, then $p_S(u) < qp_R(u)\, \forall u \in U$, and hence $1 = \sum_{u \in U} p_S(u) < q \sum_{u \in U} p_R(u) = q$, a contradiction. (b) Define $J(u) = p_S(u) - qp_R(u)$, and let $u^* = \arg\max_{u \in U} J(u)$. But $J(u) \geq 0\, \forall\, u \in \Sigma_q$, and since $\Sigma_q \neq \emptyset$, $J(u^*) \geq \max_{u \in \Sigma_q} J(u) \geq 0$, which implies $u^* \in \Sigma_q$. □

Individual rationality requires that a single ordering be defined for each agent, and that all of its options be rank-ordered with the best one surviving. This is an inter-option, or *extrinsic*, comparison, since it requires the evaluation of an option with respect to quantities other than those associated with itself (namely, ranking of all other options). The PLRT provides another way to order, using two preference orderings: one to characterize the desirable, or selectable, attributes of the options, while the other characterizes the undesirable, or rejectable, attributes, and compares these two orderings for each option, yielding a binary decision (reject or retain) for each. Such intra-option comparisons are *intrinsic*, since they do not require the evaluation of an option with respect to quantities other than those associated with itself. This intrinsic comparison identifies all options for which the benefit derived from implementing them is at least as great as the cost incurred. This notion of satisficing is compatible with Simon's original notion in that it addresses exactly the same issue that motivated Simon—to identify options that are good enough by directly comparing attributes of options. This notion differs only in the standard used for comparison. The standard for satisficing *à la* Simon, as with individual rationality in general, is imposed from without—it is extrinsic, since it relies upon external information (the aspiration level). In contrast, the standard for satisficing *à la* the PLRT is set up from within—it is intrinsic, and compares the positive attributes to the negative attributes of each option.

Intrinsic satisficing may be blended with Simon's extrinsic approach by specifying the aspiration level via the PLRT, rather than a fixed threshold. Searching then may stop when the first element of Σ_q is identified. On the other hand, searching may continue to exhaustion, and additional ordering constraints can be imposed on the elements of Σ_q to identify an optimal solution (for example, see [26]).

4.6.3 Extension to Multiple Agents

Individual satisficing is defined in terms of univariate selectability and rejectability mass functions that provide separate orderings for success and resource consumption, respectively. Just as univariate probability theory extends to multivariate probability theory, we may extend single-agent selectability and rejectability mass functions to the multi-agent case by defining a multi-agent (joint) selectability mass function to characterize group selectability and a joint rejectability function to characterize group rejectability. Given such functions, we may define a concept of multi-agent satisficing, or jointly satisficing, as follows:

Definition 4.8 A decision-making group is *jointly satisficingly rational* if the members of the group choose a vector of options for which joint selectability is greater than or equal to the index or caution times joint rejectability. □

For this definition to be useful we must be able to construct the joint selectability and rejectability functions in a way that accommodates partial preference orderings and conditional preferences. To establish this utility, we first introduce the notion of interdependence and define a satisficing game. We then describe how the interdependence function can be constructed from local orderings, leading to emergent total preference orderings.

Interdependence

An act by any member of a multi-agent system has possible ramifications throughout the entire community. Some agents may be benefited by the act, some may be damaged, and some may be unaffected. Furthermore, although the single agent may perform the act in its own interest, or for the benefit (or detriment) of other agents, the act is usually not implemented free of cost. Resources are expended, or risk is taken, or some other cost, penalty, or unpleasant consequence is incurred by the agent itself or by other agents. Although these undesirable consequences may be defined independently from the benefits, the measures associated with benefits and costs cannot be specified independently of each others due to the possibility of interaction. A critical aspect of modeling the behavior of such a society, therefore, is the means of representing the interdependence of both positive and negative consequences of all possible joint actions that could be undertaken.

Definition 4.9 Let $\{X_1, \ldots, X_N\}$ be an N-member multi-agent system. A *mixture*[5] is any subset of agents considered in terms of their interaction with each other, exclusively of possible interactions with other agents not in the subset.

A *selectability mixture,* denoted $\mathcal{S} = S_{i_1} \ldots S_{i_k}$, is a mixture consisting of agents $X_{i_1}, \ldots X_{i_k}$ being considered from the point of view of success. The *joint selectability mixture* is the selectability mixture consisting of all agents in the system, denoted $\mathbf{S} = S_1 \ldots S_N$.

A *rejectability mixture*, denoted $\mathcal{R} = R_{j_1} \ldots R_{j_\ell}$, is a mixture consisting of agents $X_{j_1}, \ldots X_{j_\ell}$ being considered from the point of view of resource consumption. The *joint rejectability mixture* is the rejectability mixture consisting of all agents in the system, denoted $\mathbf{R} = R_1 \ldots R_N$.

An *intermixture* is the concatenation of a selectability mixture and a rejectability mixture, and is denoted $\mathcal{SR} = S_{i_1} \ldots S_{i_k} R_{j_1} \ldots R_{j_\ell}$. The *joint intermixture* is the concatenation of the joint selectability and joint rejectability mixtures, and is denoted $\mathbf{SR} = S_1 \ldots S_N R_1 \ldots R_N$. □

Definition 4.10 Let U_i be the action space for X_i, $i = 1, \ldots, N$. The *product action space,* denoted $\mathbf{U} = U_1 \times \cdots \times U_N$ is the set of all N-tuples $\mathbf{u} = (u_1, \ldots, u_N)$ where $u_i \in U_i$. The *selectability action space* associated with a selectability mixture $\mathcal{S} = S_{i_1} \ldots S_{i_k}$ is the product space $\mathbf{U}_{\mathcal{S}} = U_{i_1} \times \cdots \times U_{i_k}$. The *rejectability action space* associated with a rejectability mixture $\mathcal{R} = R_{j_1} \ldots R_{j_\ell}$ is the product space $\mathbf{U}_{\mathcal{R}} = U_{j_1} \times \cdots \times U_{j_\ell}$. The *interaction space* associated with an intermixture $\mathcal{SR} = S_{i_1} \cdots S_{i_k} R_{j_1} \cdots R_{j_\ell}$ is the product space $\mathbf{U}_{\mathcal{SR}} = \mathbf{U}_{\mathcal{S}} \times \mathbf{U}_{\mathcal{R}} = U_{i_1} \times \cdots \times U_{i_k} \times U_{j_1} \times \cdots \times U_{j_\ell}$. The *joint interaction space* is $\mathbf{U}_{\mathbf{SR}} = \mathbf{U} \times \mathbf{U}$. □

[5] Not to be confused with a mixture of distributions, which is a convex combination of probability distributions.

Definition 4.11 A *selectability mass function* (smf) for the mixture $\mathcal{S} = \{S_{i_1} \dots S_{i_k}\}$ is a mass function denoted $p_{\mathcal{S}} = p_{S_{i_1},\dots,S_{i_k}}\colon \mathbf{U}_{\mathcal{S}} \to [0,1]$. The *joint smf* is an smf for $\mathbf{S}$, denoted $p_{\mathbf{S}}$.

A *rejectability mass function* (rmf) for the mixture $\mathcal{R} = \{R_{j_1} \dots, R_{j_\ell}\}$ is a mass function denoted $p_{\mathcal{R}} = p_{R_{j_1},\dots,R_{j_\ell}}\colon \mathbf{U}_{\mathcal{R}} \to [0,1]$. The *joint rmf* is a rmf for $\mathbf{R}$, denoted $p_{\mathbf{R}}$.

An *interdependence mass function* (IMF) for the intermixture $\mathcal{SR} = \{S_{i_1} \dots S_{i_k} R_{j_1} \dots R_{j_\ell}\}$ is a mass function denoted $p_{\mathcal{SR}} = p_{S_{i_1},\dots,S_{i_k},R_{j_1},\dots,R_{j_\ell}}\colon \mathbf{U}_{\mathcal{S}} \times \mathbf{U}_{\mathcal{R}} \to [0,1]$. The *joint IMF* is an IMF for $\mathbf{SR}$, denoted $p_{\mathbf{SR}}$. □

Let $\mathbf{v} \in \mathbf{U}_{\mathcal{S}}$ and $\mathbf{w} \in \mathbf{U}_{\mathcal{R}}$ be two option vectors. Then $p_{\mathcal{S},\mathcal{R}}(\mathbf{v},\mathbf{w})$ is a representation of the success support associated with $\mathbf{v}$ and the resource consumption associated with $\mathbf{w}$ when the two option vectors are viewed simultaneously. In other words, $p_{\mathcal{S},\mathcal{R}}(\mathbf{v},\mathbf{w})$ is the mass associated with selecting $\mathbf{v}$ in the interest of success and rejecting $\mathbf{w}$ in the interest of conserving resources.

Satisficing Games

The interdependence function incorporates all of the information relevant to the multi-agent decision problem. From this function we may derive the joint selectability and rejectability marginals as

$$p_{\mathbf{S}}(\mathbf{u}) = \sum_{\mathbf{v}\in\mathbf{U}} p_{\mathbf{S},\mathbf{R}}(\mathbf{u},\mathbf{v}) \tag{4.9}$$

$$p_{\mathbf{R}}(\mathbf{v}) = \sum_{\mathbf{u}\in\mathbf{U}} p_{\mathbf{S},\mathbf{R}}(\mathbf{u},\mathbf{v}) \tag{4.10}$$

for all $(\mathbf{u},\mathbf{v}) \in \mathbf{U} \times \mathbf{U}$. Once these quantities are in place, a satisficing game can be formally defined.

Definition 4.12 A *satisficing game* for a set of decision makers $\{X_1,\dots,X_N\}$, is a triple $\{\mathbf{U}, p_{\mathbf{S}}, p_{\mathbf{R}}\}$, where $\mathbf{U}$ is a joint action space, $p_{\mathbf{S}}$ is the joint selectability function, and $p_{\mathbf{R}}$ is the joint rejectability function. The *joint solution* to a satisficing game with index of caution q is the set

$$\mathbf{\Sigma}_q = \{\mathbf{u} \in \mathbf{U}\colon p_{\mathbf{S}}(\mathbf{u}) \geq q p_{\mathbf{R}}(\mathbf{u})\}. \tag{4.11}$$

$\mathbf{\Sigma}_q$ is termed the *joint satisficing* set, and elements of $\mathbf{\Sigma}_q$ are *jointly satisficing actions*. Equation (4.11) is the *joint praxeic likelihood ratio test* (JPLRT). □

The JPLRT establishes group preferences and identifies the joint option vectors that are satisficing from the group perspective. The marginal selectability and rejectability mass functions for each X_i may be obtained from (4.9) and (4.10), yielding:

$$p_{S_i}(u_i) = \sum_{\substack{u_j \in U_j \\ j \neq i}} p_{S_1,\dots,S_N}(u_1,\dots,u_N) \tag{4.12}$$

$$p_{R_i}(u_i) = \sum_{\substack{u_j \in U_j \\ j \neq i}} p_{R_1,\dots,R_N}(u_1,\dots,u_N). \tag{4.13}$$

Definition 4.13 The *individual solutions* to the satisficing game $\{\mathbf{U}, p_{\mathbf{S}}, p_{\mathbf{R}}\}$ are the sets

$$\Sigma_q^i = \{u_i \in U_i : p_{S_i}(u_i) \geq q p_{R_i}(u_i)\}, \tag{4.14}$$

where p_{S_i} and p_{R_i} are given by (4.12) and (4.13), respectively, for $i = 1, \ldots, N$. The product of the individually satisficing sets is the *satisficing rectangle*:

$$\mathfrak{R}_q = \Sigma_q^1 \times \cdots \times \Sigma_q^N = \{(u_1, \ldots, u_N) : u_i \in \Sigma_q^i\}.$$

□

It remains to determine the relationship between the jointly satisficing set $\boldsymbol{\Sigma}_q$ and the individually satisficing sets, Σ_q^i, $i = 1, \ldots, N$. Unfortunately, it is not generally true that either $\boldsymbol{\Sigma}_q \subset \mathfrak{R}_q$ or $\mathfrak{R}_q \subset \boldsymbol{\Sigma}_q$. The following result, however, is very useful.

Theorem 3 (The Negotiation Theorem) *If u_i is individually satisficing for X_i, that is, if $u_i \in \Sigma_q^i$, then it must be the ith element of some jointly satisficing vector $\mathbf{u} \in \boldsymbol{\Sigma}_q$.*

Proof This theorem is proven by establishing the contrapositive, namely, that if u_i is not the ith element of any $\mathbf{u} \in \boldsymbol{\Sigma}_q$, then $u_i \notin \Sigma_q^i$. Without loss of generality, let $i = 1$. By hypothesis, $p_{\mathbf{S}}(u_1, \mathbf{v}) < q p_{\mathbf{R}}(u_1, \mathbf{v})$ for all $\mathbf{v} \in U_2 \times \cdots \times U_N$, so $p_{S_1}(u_1) = \sum_{\mathbf{v}} p_{\mathbf{S}}(u_1, \mathbf{v}) < q \sum_{\mathbf{v}} p_{\mathbf{R}}(u_1, \mathbf{v}) = q p_{R_1}(u_1)$, hence $u_1 \notin \Sigma_q^1$. □

The content of this theorem is that no one is ever completely frozen out of a deal—every decision maker has, from its own perspective, a seat at the negotiating table. This is perhaps the weakest condition under which negotiations are possible. If $\boldsymbol{\Sigma}_q \cap \mathfrak{R}_q$ is empty, then there are no jointly satisficing options that are also individually satisficing for all players for the given value of q. The following corollary, whose proof is trivial and is omitted, addresses this situation.

Corollary 1 *There exists an index or caution value $q_0 \in [0, 1]$ such that $\boldsymbol{\Sigma}_{q_0} \cap \mathfrak{R}_{q_0} \neq \emptyset$.*

Thus, if the players are each willing to lower their standards sufficiently by decreasing the index of caution, q, they may eventually reach a compromise that is both jointly and individually satisficing, according to a reduced level of what it means to be good enough. The parameter q_0 is a measure of how much they must be willing to compromise to avoid an impasse. Note that willingness to lower one's standards is not total capitulation, since the participants are able to control the degree of compromise by setting a limit on how small of a value of q they can tolerate. Thus, a controlled amount of altruism is possible with this formulation. But, if any player's limit is reached without a mutual agreement being obtained, the game has reached an impasse.

It may be observed that the negotiation theorem does not provide for solutions which are both individually and jointly satisficing for all agents. This requires separate efforts at coordination in an active process of working toward an accord. This process is explored in [21].

Synthesis

The joint IMF provides a complete description of the individual and interagent relationships in terms of their positive and negative consequences, and provides a total ordering for both selectability and rejectability for the entire community as well as for each individual. Basing a praxeology on the IMF does not, at first glance, however, appear to conform to the requirement to accommodate partial orderings, but first glances can be misleading. Fortunately, the IMF, based as it is on the mathematics of probability theory, can draw upon a fundamental property of that theory, namely, the law of compound probability, to simplify its construction.

The law of compound probability says that joint probabilities can be constructed from conditional probabilities and marginal probabilities. For example, we may construct a joint probability mass function $p_{X,Y}(x,y)$ from the conditional mass function $p_{X|Y}(x|y)$ and the marginal $p_Y(y)$ according to Bayes rule, yielding $p_{X,Y}(x,y) = p_{X|Y}(x|y)p_Y(y)$. This relationship may be extended to the general multivariate case by repeated applications, yielding what is often termed the *chain rule*.

Definition 4.14 Given an intermixture $\mathcal{SR} = S_{i_1} \dots S_{i_k} R_{j_1} \dots R_{j_\ell}$, a *subintermixture* of $\mathcal{SR}$ is an intermixture formed by concatenating subsets of $\mathcal{S}$ and $\mathcal{R}$: $\mathcal{S}_1\mathcal{R}_1 = S_{i_{p_1}} \dots S_{i_{p_q}} R_{j_{r_1}} \dots R_{j_{r_s}}$, where $\{i_{p_1}, \dots, i_{p_q}\} \subset \{i_1, \dots, i_k\}$ and $\{j_{r_1}, \dots, j_{r_s}\} \subset \{j_1, \dots, j_\ell\}$. The notation $\mathcal{S}_1\mathcal{R}_1 \subset \mathcal{SR}$ indicates that $\mathcal{S}_1\mathcal{R}_1$ is a subintermixture of $\mathcal{SR}$.

The $\mathcal{SR}$*-complementary subintermixture* associated with a subintermixture $\mathcal{S}_1\mathcal{R}_1$ of an intermixture $\mathcal{SR}$, denoted $\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1$, is an intermixture created by concatenating the selectability and rejectability mixtures formed by the relative compliments of $\mathcal{S}_1$ and $\mathcal{R}_1$. Clearly, $\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1 \subset \mathcal{SR}$. $\mathcal{SR}$ is the union of $\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1$ and $\mathcal{S}_1\mathcal{R}_1$, denoted $\mathcal{SR} = \mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1 \cup \mathcal{S}_1\mathcal{R}_1$. □

Definition 4.15 Let $\mathcal{SR}$ be an intermixture with subintermixture $\mathcal{S}_1\mathcal{R}_1$. A *conditional interdependence mass function*, denoted $p_{\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1|\mathcal{S}_1\mathcal{R}_1}$, is a mapping of $(\mathbf{U}_{\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1} \times \mathbf{U}_{\mathcal{S}_1\mathcal{R}_1})$ into $[0,1]$ such that, for every $\mathbf{v} \in \mathbf{U}_{\mathcal{S}_1\mathcal{R}_1}$, $p_{\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1|\mathcal{S}_1\mathcal{R}_1}(\cdot|\mathbf{v})$ is a mass function on $\mathbf{U}_{\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1}$. □

All conditional interdependence mass functions must be be consistent with interdependence mass functions. That is, for $\mathcal{SR}$ an arbitrary intermixture with subintermixture $\mathcal{S}_1\mathcal{R}_1$ with $\mathbf{w} \in \mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1$ and $\mathbf{v} \in \mathcal{S}_1\mathcal{R}_1$, Bayes rule requires that

$$p_{\mathcal{S},\mathcal{R}}(\mathbf{v},\mathbf{w}) = p_{\mathcal{SR}\backslash\mathcal{S}_1\mathcal{R}_1|\mathcal{S}_1\mathcal{R}_1}(\mathbf{w}|\mathbf{v}) \cdot p_{\mathcal{S}_1\mathcal{R}_1}(\mathbf{v}). \tag{4.15}$$

This is the chain rule applied to intermixtures. Repeated applications of the chain rule provides a way to construct global behavior from local behavioral relationships. To illustrate, let $\{X_1, X_2, X_3\}$ be a multi-agent system and let $\mathcal{S} = S_1S_2$ and $\mathcal{R} = R_3$. Then $\mathcal{SR} = S_1S_2R_3$ and $\mathbf{SR}\backslash\mathcal{SR} = S_3R_1R_2$. The IMF is

$$p_{S_1,S_2,S_3,R_1,R_2,R_3}(v_1,v_2,v_3,w_1,w_2,w_3) = \\ p_{S_3,R_1,R_2|S_1,S_2,R_3}(v_3,w_1,w_2|v_1,v_2,w_3) \cdot p_{S_1,S_2,R_3}(v_1,v_2,w_3).$$

Now let $\mathcal{S}_1 = S_1$ be a subintermixture of $S_1S_2R_3$, so that $\mathcal{SR}\backslash\mathcal{S}_1 = S_2R_3$. We may apply the chain rule to this subintermixture to obtain

$$p_{S_1,S_2,R_3}(v_1,v_2,w_3) = p_{S_1|S_2,R_3}(v_1|v_2,w_3) \cdot p_{S_2,R_3}(v_2,w_3),$$

yielding

$$p_{S_1,S_2,S_3,R_1,R_2,R_3}(v_1,v_2,v_3,w_1,w_2,w_3) = p_{S_3,R_1,R_2|S_1,S_2,R_3}(v_3,w_1,w_2|v_1,v_2,w_3) \cdot p_{S_1|S_2,R_3}(v_1|v_2,w_3) \cdot p_{S_2,R_3}(v_2,w_3). \quad (4.16)$$

The term $p_{S_3,R_1,R_2|S_1,S_2,R_3}(v_3,w_1,w_2|v_1,v_2,w_3)$ is the conditional selectability/rejectability associated with X_3 selecting v_3, X_1 rejecting w_1, and X_2 rejecting w_2, given that X_1 prefers to select v_1, X_2 prefers to select v_2, and X_3 prefers to reject w_3; $p_{S_1|S_2,R_3}(v_1|v_2,w_3)$ characterizes X_1's selectability for v_1 given X_2 prefers to select v_2 and X_3 prefers to reject w_3; and $p_{S_2,R_3}(v_2,w_3)$ is the joint selectability/rejectability of X_2 selecting v_2 and X_3 rejecting w_3. The various terms of this factorization may often be simplified further. For example, suppose that X_1 is indifferent to X_3's rejectability posture, in which case we may simplify $p_{S_1|S_2,R_3}(v_1|v_2,w_3)$ to become $p_{S_1|S_2}(v_1|v_2)$.

Clearly, there are many ways to factor the interdependence function according to the chain rule. The design issue, however, is to implement a factorization that allows the desired local interdependencies to be expressed through the appropriate conditional interdependencies. The construction of the interdependence function is highly application dependent, and there is no general algorithm or procedure that a designer should follow for its synthesis. There are, however, some general guidelines for the construction of interdependence functions.

1. Form operational definitions of selectability and rejectability for individuals or groups, as appropriate from the context of the problem.

2. Identify the local orderings that are desirable, and map these into conditional selectability and rejectability functions.

3. Factor the interdependence function such that the desired conditional selectability/rejectability relationships are products in the factorization.

4. Eliminate all irrelevant interdependencies in the factors.

Meso-Emergence

Although each of the conditional mass functions in the factorization of the interdependence function is a total ordering, it is a *local* total ordering, and involves only a subset of agents and concerns. Each of these local total orderings is only a partial ordering, however, if viewed from the global, or community-wide, perspective, since orderings are not defined for all possible option vectors. By combining such local total orderings together according to the chain rule, a global total ordering emerges. The joint selectability and rejectability mass functions then characterize emergent global behavior, and the individual selectability and rejectability marginals characterize emergent individual behavior. Thus, both individual and group behavior emerge as consequences of local conditional interests that propagate throughout the community from the interdependent local to the interdependent global and from the conditional to the unconditional.

Synthesizing the IMF exploits an emergence property that is quite different from the temporal, or evolutionary, emergence that can occur with repeated play games. To differentiate these two types of emergence, let us refer to the former as *spatial* emergence. Temporal emergence is an inter-game phenomenon that produces relationships between agents with repeated play as time propagates, and spatial emergence is an intra-game phenomenon that produces relationships between agents as interests propagate through the agent system with single-play. Perhaps the most common example of spatial emergence is the *micro-to-macro*, or *bottom-up* phenomenon of group behavior emerging as a consequence of individual interests, as occurs with social choice theory [83, 40] and with evolutionary games [84, 85]. A second approach is a macro-to-micro or *top-down* approach, where individual behaviors emerge as a consequence of group interests. Satisficing praxeology accommodates both of these approaches. It also points to a third approach, that of an *inside-out*, or meso-to-micro/macro view, where intermediate-level conditional preferences propagate up to the group level and down to the individual level. Let us term this type of spatial emergence *meso-emergence*.

The conditional selectability and rejectability mass functions are constructed as functions of the preferences of the other agents. For example, the local total ordering function $p_{S_1|S_2}(\cdot|v_2)$ characterizes X_1's ordering of its selectability preferences given that X_2 prefers v_2. This structure permits X_1 to ascribe some weight to X_2's interests without requiring X_1 to abandon its own interests in deference to X_2. By adjusting these weights, X_1 may control the degree two which it is willing to compromise its egoistic values to accommodate X_2.

4.6.4 Discussion

The group decision problem has perplexed researchers for decades. As [86, pp. 233–237] put it over thirty years ago, "I find myself in that uncomfortable position in which the more I think the more confused I become." The source of Raiffa's concern, it seems, is that it is difficult to reconcile the notion of individual rationality with the belief that "somehow the group entity is more than the totality of its members." Yet, researchers have steadfastly and justifiably refused to consider the group entity itself as a decision-making superplayer.

Satisficing game theory offers a way to account for the group entity without the fabrication of a superplayer. This accounting is done through the conditional relationships that are expressed through the interdependence function due to its mathematical structure as a probability (but not with the usual semantics of randomness). Just as the a joint probability function is more than the totality of the marginals, the interdependence function is more than the totality of the individual selectability and rejectability functions. It is only in the case of stochastic independence that a joint distribution can be constructed from the marginal distributions, and it is only in the case of complete inter-independence that group welfare can be expressed in terms of the welfare of the individuals.

The current literature on negotiation concentrates heavily on ways to obtain just-in-time negotiated solutions that can be accomplished within real-time computational constraints, but it does so *primarily from the point of view of individual rationality*. There is no reason, however, to limit consideration to that perspective. This paper is an invitation to expand to a broader perspective, and consider dealing with the exigencies of practical decision making in the light of satisficing game theory as well as with conventional theory.

Negotiation under (bounded or unbounded) rational choice requires the decision maker to attempt to maximize its own benefit. This is a valid, and perhaps the only reliable, paradigm in extremely conflictive environments, such as zero-sum games, but when the opportunity for cooperation exists, the rational choice paradigm is overly pessimistic and unnecessarily limits the scope of negotiation.

The appeal of optimization, no matter now approximate, is a strongly entrenched attitude that dominates current decision making practice. There is great comfort in following traditional paths, especially when those paths are founded on such a rich and enduring tradition as rational choice affords. But when synthesizing an artificial negotiatory system, the designer has the opportunity to impose upon the agents a more socially accommodating paradigm. The satisficing game theory presented in this paper provides a sociological decision-making mechanism that seamlessly accounts for group and individual interests, and provides a rich framework for negotiation to occur between agents who share common interests and who are willing to give deference to each other. Rather than depending upon the non-cooperative equilibria defined (even if only approximately) by individual-benefit saddle points, this alternative may lead to the more socially realistic and valuable equilibria of shared interests and acceptable compromises.

4.7 A Market Approach to Coordination

In this section, we present concepts that attempt to relate the praxeological approach with market dynamics. The market concepts are derived following ideas of [87].

An economy consisting of needs and abilities to satisfy those needs. Perhaps we can characterize both as "goods". Some agents will bring as their goods an excess of needs, which they desire to trade for the abilities of someone else. Others will bring an excess of abilities, which they desire to trade for the needs of someone else. Somehow price needs to fit into all of this.

In the economy of interest here, we envision two classes of "goods." Let $\mathcal{D} = \{d_1, d_2, \ldots, d_{n_1}\}$ denote a class of *demands* which may be brought to bear by some agents, and let $\mathcal{S} = \{s_1, s_2, \ldots, s_{n_2}\}$ denote a class of *supplies* which may be brought to bear by some agents. There are thus $n = n_1 + n_2$ types of goods.

In the economy of supplies and demands, demands can be met by certain equivalences in supplies. For example, we might have

$$d_1 = 2s_1 + 3s_2,$$

so that a single unit of demand of type d_1 is met by 2 units of supply s_1 and 3 units of supply s_2. We assume that there is a linear constituitive relationship

$$\mathbf{d} = B\mathbf{s}.$$

Assume that there are M agents in the system. Agent i is provided with the allocation of goods $\mathbf{w}^i = [w_1^i, w_2^i, \ldots, w_n^i]^T$, where w_j^i is the amounts of good j possessed by agent i.

Each agent is also provided with a utility function $f^i(\mathbf{w}^i)\colon \mathbb{R}^n \to \mathbb{R}$. (As development proceeds, we may want to include both selectability and rejectability utility functions.) The goal is to enter into market negotiations so that agents are able to increase their utility by exchanging their own endowment of goods $\mathbf{w}^i$ for another endowment of goods. Let $\mathbf{x}^i = [x_1^i, x_2^i, \ldots, x_n^i]^T$ be the vector of goods of agent i after a round of market trading.

In this economy, there is also a price vector $\mathbf{p} = [p_1, p_2, \ldots, p_n]^T$, arrived at largely by market forces, which determines how goods are traded. The amount that agent i stands to receive for his allotment of goods is

$$m^i = \sum_{j=1}^{n} p_j w_j^i = \mathbf{p}^T \mathbf{w}^i.$$

Thus m^i is the amount of *budget* available to agent i for dealings in the market. [We may also find it convenient to endow each agent with some another source of budget which is not tied to any good, i.e., money. This could be used to give them greater flexibility in the market.] To keep within budget, we must have

$$\mathbf{p}^T \mathbf{x}^i \leq \mathbf{p}^T \mathbf{w}^i$$

(money spent does not exceed money available).

Let $w_j = \sum_{i=1}^{M} w_j^i$ be the total amount of good j available in the market among all agents, and let $\mathbf{w} = [w_1, w_2, \ldots, w_n]^T$. Then, since an agent cannot purchase more than is available, we must have

$$0 \leq \mathbf{x}^i \leq \mathbf{w}.$$

Let $\mathbf{x} = [\mathbf{x}^1; \mathbf{x}^2; \ldots; \mathbf{x}^M] \in \mathbb{R}^{Mn}$ denote the total vector of goods after market. The market determines the vector $[\mathbf{x}; \mathbf{p}]^T \in \mathbb{R}^{(M+1)n}$.

Under a state of competitive equilibrium, everyone is satisfied to some degree. The problem of competitive equilibrium can be stated as:

For $i = 1, 2, \ldots, M$, for a price vector $\mathbf{p} = \overline{\mathbf{p}}$, find $\overline{\mathbf{x}}^i$ to satisfy

$$f^i(\overline{\mathbf{x}}^i) = \max_{\overline{\mathbf{x}}} f^i(\overline{\mathbf{x}}) \tag{4.17}$$

subject to

$$\overline{\mathbf{p}}^T \overline{\mathbf{x}}^i \leq \overline{\mathbf{p}}^T \mathbf{w}^i \tag{4.18}$$

$$0 \leq \overline{\mathbf{x}}^i \leq \mathbf{w} \tag{4.19}$$

and

$$\sum_{i=1}^{M} \overline{\mathbf{x}}^i \leq \mathbf{w} \tag{4.20}$$

$$\overline{\mathbf{p}} \geq \mathbf{0} \qquad \sum_{j=1}^{n} \overline{p}_j = 1. \tag{4.21}$$

In this economy, agents whose goods are primarily among $\mathcal{D}$ are termed *demanders*, and those whose goods are primarily among $\mathcal{S}$ are called suppliers. Demanders have utility functions that favor small values of the goods they have in demand. In essence, they sell their demands to suppliers. Suppliers, buy the demands by selling their supplies, and have utility functions which favor small values of their supplies.

[There seems to be a problem in the pricing structure as stated so far. I haven't yet tied in the constituitive relationship, nor excluded the possibility of an agent selling his demands to another agent for demands. Also, the notion of time to completion is not yet entered in as an explicit part of the model.]

Example 4.7.1 *Woofers (d_1) and weefers (d_2) are made out of widgets (s_1) and wadgets (s_2) according to the formula*

$$d_1 = 2s_1 + 3s_2$$

$$d_2 = 4s_1 + s_2$$

Let Adam (X_1), Eve (X_2), Cain (X_3) and Abel (X_4) be four agents, where Adam and Eve are demanders, and Cain and Abel are suppliers. The initial allocation to these agents is

$$\mathbf{w}^1 = (2, 2, 0, 0)$$

(Adam wants 2 woofers and 2 weefers, and starts with no supplies)

$$\mathbf{w}^2 = (4, 5, 0, 0)$$

(Eve wants 4 woofers and 5 weefers, and starts with no supplies)

$$\mathbf{w}^3 = (0, 0, 4, 4)$$

(Cain can supply 4 each of widgets and wadgets) and

$$\mathbf{w}^4 = (0, 0, 3, 3)$$

(Abel can supply 3 each of widgets and wadgets). Clearly not everyone is going to be fully satisfied since there are insufficient materials.

The utility functions are

$$f^1(\mathbf{x}^1) = -(x_1^1)^2 - (x_2^1)^2$$

(Adam wants to drive down the number of demands he has, by selling them off).

Or perhaps what we want is to incorporate the consituitive relationships right from the beginning: we want to drive the demands down to zero, by increasing the corresponding supplies. Maybe we should have

$$f^1(\mathbf{x}^1) = -(x_1^1)^2 - (x_2^1)^2 + [x_1^1(2x_3^1 + x_4^1)]^2 + [x_2^1(4x_3^1 + 3x_4^1)]^2,$$

and similarly for the other functions.

$$f^2(\mathbf{x}^2) = -10(x_1^2)^2 - 5(x_2^2)^2$$

(Eve wants to drive down the number of demands she has, by selling them off, but appears to be more demanding; should we call her Lillith?)

$$f^3(\mathbf{x}^3) = -(x_3^3)^2 - (x_4^3)^2$$

(Cain is happy to get rid of his supplies)

$$f^4(\mathbf{x}^4) = -(x_3^4)^2 - (x_4^4)^2$$

(Abel is happy to get rid of his supplies). □

4.7.1 Ant pile

An interesting and potentially rich problem for study: There is a pile of resource — "food" — which is sought by M teams of agents. We will use the abbreviation "ants" for these agents. The goal of the game is for each team to transport as much of the food as possible to their base. Members of the teams may be endowed with different physical capabilities, such as carriers, blockers, guards, etc., and with varying cognitive abilities. This therefore may be considered a generalization of the capture the flag game studied by Goodrich [88, page 72]. Rather than having a single flag, the food resource may be viewed as a large heap of flags. Thus, a single game provides more opporunity for dynamics to develop and continue than in single-flag flag play, and the results of the game may better reflect ensemble properties of play. A variety of variations on the basic game are possible, such as:

- The location of the food may be initially unknown, reguiring some mapping capability. Or the location may change from time to time.
- The size of the pile of food may vary physically, becoming smaller as food is depleted from it, and hence requiring more travel time.
- The ants may have various sensory limitations placed upon them. For example, they may play at night, being only able to communicate and locate by touch.
- Coalition play may be possible, where the winning score is to some coalition of teams.
- Time deadlines may be imposed. In this mode, even single-team play becomes interesting, as the team organize must move the maximum amount of food in the given time.
- Modeling details such as incorporating the cost of deliberation/negotiation may be included.
- Strength abilities of an ant may also be interesting to model, so that ants may carry different loads.

The overall goal of the problem is to provide a framework in which interesting negotiation can take place, then to look for principles of negotiation dynamics which may have general applicability.

4.7.2 Ant postures

We model the behavior dynamic of an ant by assuming that an ant may assume different postures at different postures. These postures include elements of the set

$$\mathcal{P} = \{\text{foodward,homeward,passer,defender,blocker,attacker}\}.$$

(Other possibilities may also arise.) These describe the following aspects of behavior.

foodward A foodward ant is moving toward generally toward the food, with possible deviations to avoid ants either on its own or other teams.

homeward A homeward ant is moving generally toward its home base (with possible deviations to avoid ants either on its own or other teams.

passer A passer is an ant who passes his food to another ant (capable of accepting it). This behavior may lead to daisy-chaining as a means of food transport.

defender A defender is an ant who defends other ants on his own team, making it possible for them to either carry food or to reach the food.

blocker A blocker ant attempts to impede the progress of an ant from another team toward its goal, either food or home.

attacker An attacker ant takes a more agressive role of actually attacking an ant (rather than just blocking it).

Extending the role of attacker, it might be interesting to incorporate an ability to steal food, perhaps if several attackers surround an ant carrying food.

Subset postures — assuming more than one of these elemental roles — may also be possible.

Which posture to an ant should assume is a itself a collective decision problem.

It may be interesting to model the carrying of the food as a modification of the mass: a food-carrying agent can't move as fast.

Another modification might be tiredness: the longer an ant moves carrying a load, the smaller the range of forces they can apply to movement. (This would tend to motivate the idea of passing on to another, fresher, ant.)

4.7.3 Some selectability and rejectability functions

Selectability should describe an agent's purpose, or objective. Rejectability should describe the penalty or cost of control.

I will now try to establish some reasonable selectability and rejectability functions for various postures.

foodward The selectabilitiy is simply a function of the distance from the agent to (the estimated location of) the food.

The rejectability is based on a desire to both evade blockers and attackers, avoid moving into a location where there is another agent, and to conserve fuel.

homeward The selectability is simply a function of the distance from the agent to "home". Alternatively, if a passer with more ability is sufficiently close, a homeward agent may choose to transfer its load to the passer (this may depend on other aspects of the passer, such as how free it is from attackers and blockers).

The rejectability is based on a desire to evade blockers and attackers, avoid moving into a location where there is another agent, and to conserve fuel.

passer A passer not carrying food has selectability based on getting food from an agent carrying food. (Of course, there must be some coordination, willingness on the part of the agent with food to transfer it to the passer agent.)

The rejectability is based on a desire to evade blockers and attackers, avoid moving into a location where there is another agent, and to conserve fuel.

defender A defender has as selectable choices those that place it between agents of the opposing team that may block or attack and members of its own team. More effectively, a defender should incorporate dynamic models of members of both teams. Also, consideration should be given (as the model develops) to obtain coordinated behavior among blockers and attackers.

The rejectability is based upon a desire to conserve fuel and avoid squares where other agents are.

blocker A blocker has as selectable choices those that place them in the path of a foodward or homeward agent, or that can block a blocker from accomplishing its task.

attacker An attacker wants to attack agents of the other team, incapacitating them.

4.7.4 Knowledge corpi

We may also want to explore various endowments of knowledge upon the agents, and determine behaviors as a function of how much they know. Here are some possiblities:

- Full knowledge: every ant on every team has knowledge of the state of every ant on every other team.
- Full knowledge + anticipation: every ant on every team has knowledge of the state of every ant on every other team, plus has the ability to predict something about the state of ants over some horizon into the future.
- Local knowledge: ants are aware only of those other ants in some neighborhood around them. Anticipation may also be incorporated.

Also, various models of command structure may be explored. For example, a top-down structure in which a single ant directs a team might be employed, or a fully distributed structure. It may even be possible to explore changing the structure based on another decision.

4.7.5 Some initial notation

We now establish some notation for this game. The ith team is denoted $\mathcal{T}_i$,[6] and the number of members on $\mathcal{T}_i$ is n_i. Ant j on team $\mathcal{T}_i$ is denoted as $\mathcal{T}_i(j)$.

[6] This differs from the notation in [88], where $\mathcal{K}$ is used to denote a coalition. However, since coalitions may be built using members of teams, we introduce notation to distinguish teams from coalitions.

Let $\mathbf{x}_{\mathcal{T}_i(j)}(t)$ denote the dynamical state of ant $\mathcal{T}_i(j)$ at time t. The dynamics are governed by the general equation

$$\mathbf{x}_{\mathcal{T}_i(j)}(t+1) = \mathbf{f}_{\mathcal{T}_i(j)}(\mathbf{x}_{\mathcal{T}_i(j)}(t), \mathbf{u}_{\mathcal{T}_i(j)}(t), t)$$

where $\mathbf{u}_{\mathcal{T}_i(j)}(t)$ is an input function at time t. To avoid the multiple subscripts, we will also employ a notation such as $\mathbf{x}_i$, where the team and member identification are implicit. Thus xbf_i and $\mathbf{x}_j$, $i \neq j$, refer to the state of two different ants. For notational brevity, we will also let $\mathbf{x}(t)$ denote the state for a general agent. Assuming that play occurs on a two-dimensional space with coordinate (x, y), a state vector for Newtonian dynamics is

$$\mathbf{x}(t) = \begin{bmatrix} x(t) \\ \mathbf{y}(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{p}(t) \\ \mathbf{v}(t) \end{bmatrix}$$

and the state update equation can be written as

$$\mathbf{x}(t+1) = A\mathbf{x}(t) + B\mathbf{u}(t)$$

where

$$A = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ T/m & 0 \\ 0 & T/m \end{bmatrix}$$

and

$$\mathbf{u}(t) = \begin{bmatrix} F_x(t) \\ F_y(t) \end{bmatrix},$$

with m being the mass of the agent, T being the sample time, and $F_x(t)$ and $F_y(t)$ are the forces applied in the x and y directions at time t.

Let $\sigma_{\mathcal{T}_i(j)}(t)$ denote the posture of the agent at time t. Strictly speaking, this should form part of the state, so it may be convenient sometimes to form the augmented state

$$\overline{\mathbf{x}}(t) = \begin{bmatrix} \mathbf{x}(t) \\ \sigma(t) \end{bmatrix}.$$

For purposes of coordination, it is necessary to an ant i retain an estimate of the state of other ants. We will let $\hat{\mathbf{x}}_j^i(t)$ denote ant i's estimate of ant j's state (or, for the augmented state, $\hat{\overline{\mathbf{x}}}_i^j(t)$).

4.8 Qualitative structural properties for multiagent systems

In this section, a completely different perspective is sought on the multiagent problem. Rather than seeking a particular control algorithm or a particular negotiation strategy, general structural properties are sought using the techniques of catastrophe theory, or, under its more modern but less dramatic name, bifurcation theory. This theory can accomodate general arguments, and leads to some semi-qualitative results.

4.8.1 Introduction

We consider the qualitative effectiveness and structural stability of a generic multiagent system, where the structural stability is considered as the number of agents and their proximity is varied. The intent is not to obtain particular numerical answers, but to explore structural (or topological) aspects of the system as parameters vary. The tools used in this exploration derive from those of catastrophe theory [89, 90, 91, 92]. Using catastrophe theory, the phase transitions which are known to occur in problems of multiagent systems are accounted for by folds in the catastrophe manifold associated with the system.

4.8.2 Catastrophe theory

Catastrophe theory was a topic of considerable research in the 1970's, has since fallen into relative dis-use. However, the mathematical models remain valid, even if some of the applications are suspect (including, perhaps, even this one). Even in its weakest applications, catastrophe theory can provide effective metaphors to describe complex behavior, even if it does not provide a justified explanation [90, p. 128]. Catastrophe theory is well-suited to problems in the softer sciences, where underlying mechanisms are frequently not understood, or in large problems, such as multiagent systems, where collective behavior is observed but is difficult to describe using conventional localized analysis.

Catastrophe theory deals with the structural properties and qualitative nature of smooth functions parameterized by continuous sets of parameters. For example, $f(x; u, v) = x^3 + xu + v$ is a set of functions (in the variable x) with parameters u and v. In catastrophe theory, the structural stability of such parameterized functions are examined: if the parameters vary slightly, does the function retain its qualitative form? Since the functions are smooth and the parameters vary continuously, the answer is usually yes, but there are some values of the parameters at which the function undergoes a qualitative change near the critical points of the function. For example, the function $f(x; 0, 0) = x^3$ has only one repeated root (at zero), and no minima or maxima, only a point of inflection. The function $f(x; -u, 0) = x^3 - \epsilon^2$ has roots at $x = 0$ and $x = \pm\epsilon$ for the parameters values $u = \epsilon^2$ and $v = 0$ — there is now a maximum value for $x < 0$ and a minimum value for $x > 0$. Thus the structural nature of the function near $x = 0$ is changed by the change from $u = 0$ to $u = \epsilon^2 > 0$. This change in behavior is what is called a catastrophe — the word in this context refers to a sudden change in qualitative nature, as opposed to a disastrous change.

Since catastrophe theory examines *qualitative* changes in the functions, as opposed to particular numerical values, another aspect of the theory is that it only provides distinctions up to smooth changes of variables (diffeomorphisms). So, for a smooth change of variables $y = \phi(x)$, the function $f(\phi(y); u, v)$ may be examined for the same qualitative behavior near its critical points as for $f(x; u, v)$ (provided that ϕ is free from critical points in the appropriate neighborhoods). Similar changes of variables are allowed for the parameters. Frequently, a change of variables is employed to move the critical points of a given function to some convenient location (such as the origin). Thus, catastrophe theory provides for classification of the critical points of functions *up to diffeomorphisms.*

One of the key results of catastrophe theory is that all systems which exhibit structural changes having up to four parameters, can be classified (up to diffeomorphism) into one of seven canonical forms. Such systems will exhibit behavior indicative of the structural change, such as jumping, hysteresis, sensitivity to parameter changes, and unstable regions.

Catastrophe theory has been used to revisit a variety of problems in the physical, biological, and social sciences. One example which seems particularly germane is the use of catastrophe theory to account for phase transitions in thermodynamic systems. This provides a useful metaphor, as phase transitions in multiagent systems are also observed.

4.8.3 A Catastrophic Work model

We examine a model for the amount of work accomplished by a set of N agents working together. In this model, two aspects of the interagent work are presented. First is the conventional division of labor — the work is split N different ways (except for some "administrative" overhead), where individual difference in agent abilities are not accounted for. This time to complete the work is thus roughly proportional to $1/N$. There is also a term to account for the multiple interactions: N agents may interact with each other in $O(N^2)$ ways, and the interaction occurs in a way to decrease the time to completion. While this particular model can (and should) be dissected and discarded as unrealistic and too simplistic, we argue that models which similarly exhibit quadratic dependence on the number of agents (due to agent interaction) combined with some linear dependence on N in such a way that a cubic term arises, under diffeomorphic transformations will likely form the canonical cusp which appears under this model. If more complicated nonlinear models are employed, they are subject also to catastrophe theory: if they don't form this canonical cusp, another canonical catastrophe likely will be evident.

Let A represent an amount of work to be accomplished (represented in units of man-hours) by N agents. In a typical situation involving multiple workers, there is some worker overhead associated with the workers. We represent this overhead by b. There is also some additional time overhead associated with each problem, which we will denote as C. Thus, if all that accrues from the presence of multiple workers is a division of labor (with some administrative overhead) we can model the time to accomplish work A as

$$T = C + \frac{A}{N-b}.$$

However, in a multiagent scenario, there is usually some additional benefit from being able to work cooperatively. We will model the additional benefit as being quadratic in the number of workers N. Thus there is a *decrease* in time due to multiple workers of EN^2. [7] Under this assumption, the time to complete the task is thus

$$T = C + \frac{A}{N-b} - EN^2. \tag{4.22}$$

[7] A better model would exhibit quadratic behavior initially, but with some saturation effect as the number of workers increases, as in $E\tanh(fN^2)$, for appropriate model constants E and f. However, results in catastrophe theory depend only on the lower-order terms in the Taylor series, so the model still works up to diffeomorphism, provided, of course, that there is some limit to the values of N employed, since negative time to complete a task is meaningless.

This heuristic model is, of course, valid only over values of N that lead to a positive time to complete the task. In the analysis that follows, we neglect the discreteness of the number of agents (or argue that only portions of agents may be employed) so that T varies smoothly with N. For simplicity we set $C = 0$, which mathematically corresponds to a simple change of variables, but in the model may lead to negative completion times.

The model (4.22) is in many ways similar to the gas equation of van der Waals equation from thermodynamics [93],[89, p. 327]. In this analogy, our time T is analogous to thermodynamic pressure P, the amount of work A is analogous to temperature, and the number of workers is analogous to the volume V. The catastrophe theoretic interpretation of the van der Waals equation has been used to account for phase transitions between solid, liquid, gas, and fluid states of matter. Due to its similarity, we expect (4.22) to exhibit similar phase transition behavior.

The critical points of (4.22) occur where $dT/dN = 0$ and $d^2T/dN^2 = 0$, which is when

$$N = \frac{b}{3} \qquad A = -\frac{8b^3E}{27} \qquad T = \frac{b^2E}{3}.$$

We designate these values as N_c, A_c and T_c. Letting

$$T' = T/T_c \qquad N' = N/N_c \qquad A' = A/A_c$$

and $C = 0$ we change coordinates so that the critical point is at (1,1,1). By this transformation, (4.22) becomes

$$T' = -\frac{8A'}{3N' - 9} - \frac{1}{3}(N')^2$$

The coordinate system is now shifted so that the critical point is at $(0, 0, 0)$ by letting $t = T' - 1$, $n = N' - 1$, and $a = A' - 1$. This gives rise to the equation

$$n^3 + un + v \tag{4.23}$$

where

$$u = 3t \qquad v = 8a - 6t$$

The solutions to (4.23) for values of u and v, which are diffeomorphically related to T and A, over some continuous range defines a surface, known as the catastrophe manifold, and sketched in figure 4.5. Equation (4.23) is the classic equation of the *cusp* catastrophe (see, e.g., [90, p. 42] or [89, p. 78ff or p. 174ff]. The lines marked B in the parameter space denote the bifurcation set, at which the catastrophe manifold jumps from one sheet to another. Interior to those lines, the manifold exhibits three values, one of which is "unattainable" (or unstable). Outside of the bifurcation set, there is only a single sheet in the manifold.

This cusp exhibits the classic attributes of catastrophe, some of which we sketch here [90, p. 12]:

Sudden jumps As the parameter v is varied across its parameter space (such as on the line l_1 shown), the path p_1 on the cusp much jumps from one sheet of the catastrophe manifold to another sheet.

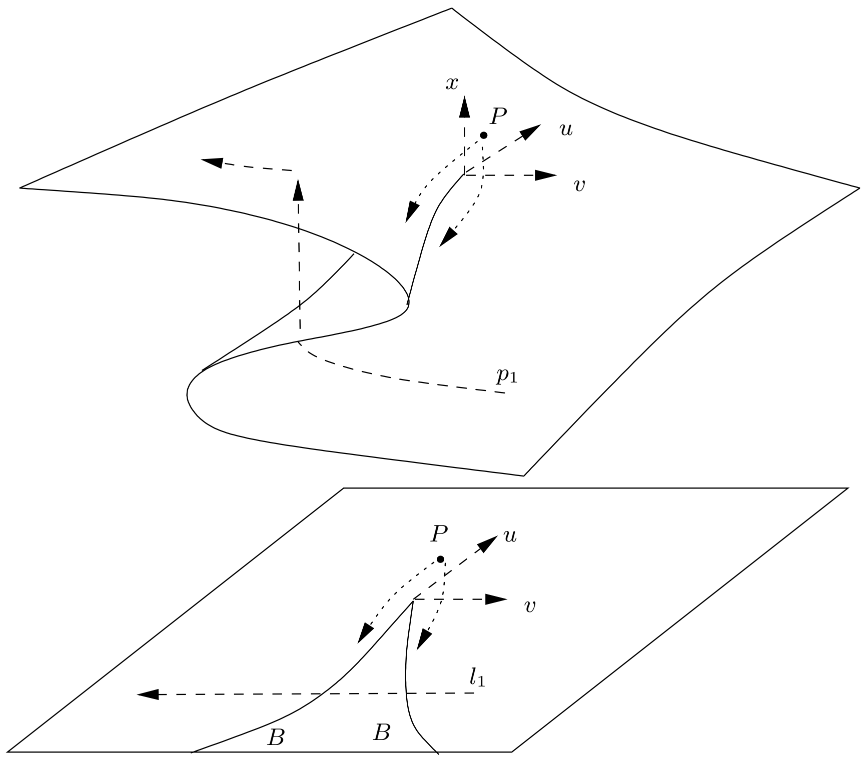


Figure 4.5: The canonical cusp catastrophe surface

Hysteresis A path in one direction across the (u, v) parameter space, followed by a return path, does not necessarily result in the same path across the folded sheet. (Think of traversing the line l_1, then traversing again in the reverse direction.)

Divergence Divergence is sensitivity to initial conditions. This is exhibited for the path starting from P and moving along similar paths in parameter space, but arriving on different sheets of the fold. Nearby trajectories in parameter space can have significantly different behavior.

The emergence of the cusp catastrophe indicates that in this multiagent model, there will be phase transitions as the parameters (amount of work or time to complete) varies, corresponding to the folds of the manifold.

4.8.4 A structural stability look at multiagent control

There is another rather different model that can be employed to describe aspects of multiagent control. We consider the "effectiveness" of a multiagent system over a distributed domain. (For a closely related model in ecology, see [89, ch. 16].) The analysis is based on some heuristic concepts of multiagent behavior:

- There is an advantage to multiagent cooperation, with per-agent effectiveness increasing as the number of agents increases. Some tasks simply require more agents to carry them through. Some tasks are intrinsically distributed. Others may benefit from a division of labor [94, p. 4].

- The benefits from coordination are limited however. Eventually saturation effects come into play, and a point of diminishing returns is reached. For example, benefits accruing from division of labor reach a saturation point when tasks can not be further subdivided. On the basis of this observation, we postulate that

Letting $S(N)$ denote the agent "effectiveness" per agent as a function of the number of agents N, we have in figure 4.6 a plot of $S(N)$ and dS/dN that represent these two heuristic concepts of multiagent behavior. The second concept can be stated as

$$\lim_{N \to \infty} \frac{dS}{dN} = 0$$

At the same time there are advantages in a multiagent system, there are also costs associated with cooperation:

- We model the cost as being related to the distance between agents. Thus, it might represent costs associated with transportation or communication.

We model the agents as being distributed over some problem domain of "size" R. The problem domain may be a geographical one, where R is a measure of the size of the region. Or, the problem domain may be a cognitive one, where R is a measure of the number of subtasks to be performed in the completion of some assigned work. We denote the "density" of the agents in the domain as

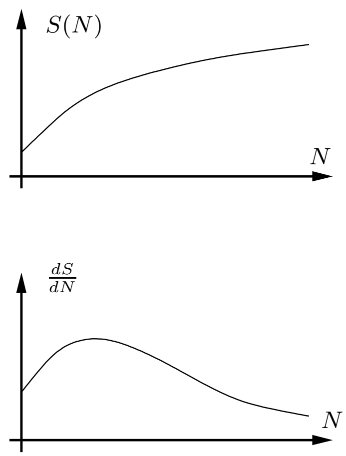


Figure 4.6: Effectiveness per agent as a function of number of agents, and its derivative

D. For a geographical domain, D represents the actual average density, as agents per unit area. For a cognitive domain, the density might be a measure of the skill level of the agents (for example, how many subtasks they are equipped to deal with). The effective number of agents N in the system with size R and density D is an increasing function of the domain size and the density. For example, for a geographical domain, we might have $N = \pi R^2 D$.

To model the cost of agent interaction, let

$$E(N) = C - d\frac{N}{D}.$$

denote how the agents' effectiveness decreases as the number of agents increases, where C and d are constants.

The total effectiveness combines the effectiveness due to agent interaction and the cost of agent interaction:

$$F(N) = E(n) + S(N) = C - dN/D + S(N)$$

Then

$$\frac{dF}{dN} = \frac{dS}{dN} - \frac{d}{D}$$

Various qualitative forms of dF/dN are obtained as d/D varies. These are shown on the left of figure 4.7. On the basis of the shape dS/dN and depending on the size of d/D, the function dF/dN may have one zero, multiple zeros, or no zeros. The corresponding total effectiveness is shown plotted next to its derivative (on the right of figure 4.7).

If the density of agents is sufficiently high — and the task distribution is such that each agent is kept productively occupied, then for high density the effectiveness is as shown in figure 4.7(b) — there is a unique value of N maximizing the effectiveness. Beyond that point, diminishing returns reduce effectiveness. As the density of productively occupied agents decreases, in figure 4.7(e) and (g), the position of most effectiveness first decreases, then it is more effective to work in isolation figure 4.7(h). These function values can be viewed as being slices from some catastrophe manifold.

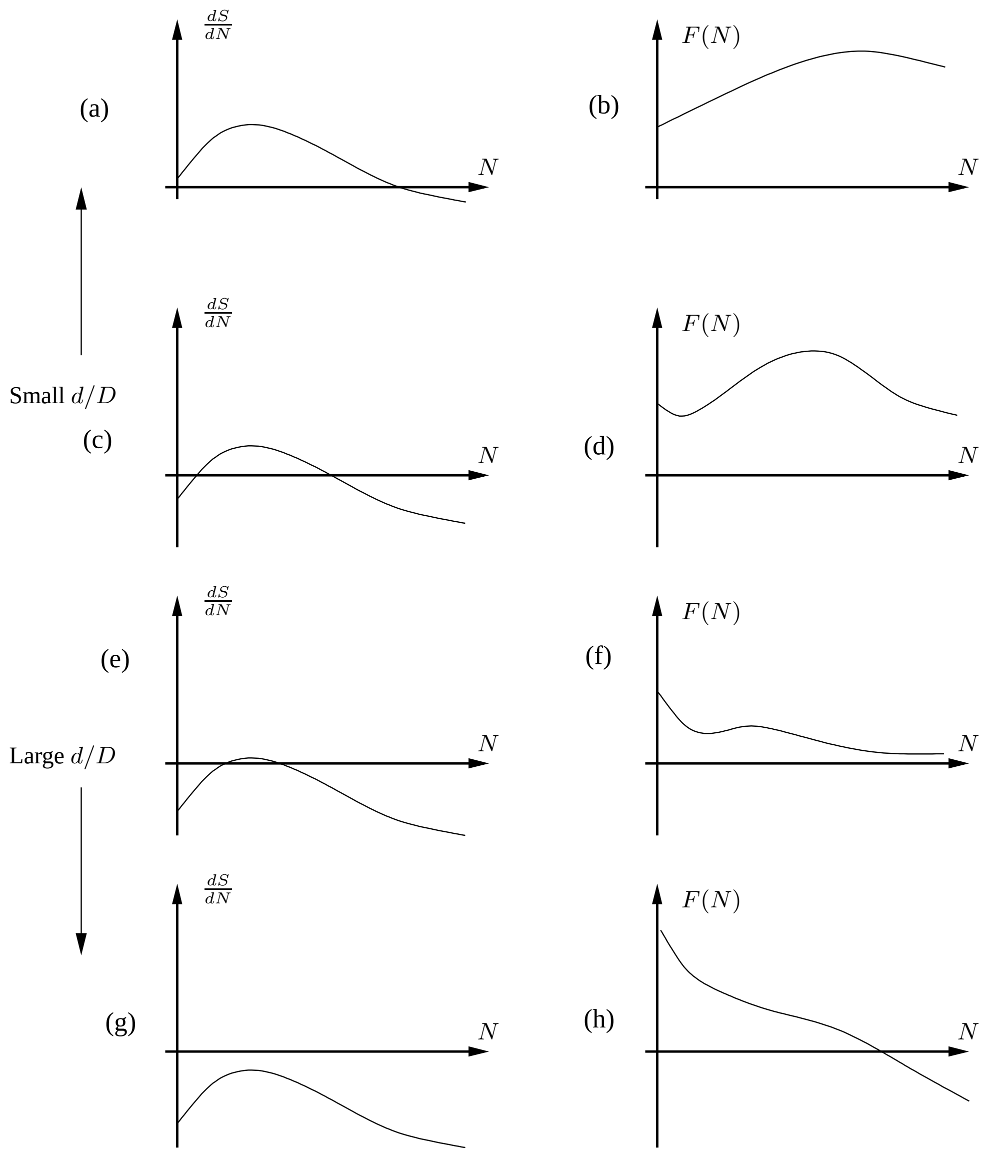


Figure 4.7: The derivative of the effectiveness, and the effectiveness, as a function of d/D

Figure 4.8.4 illustrates the plot of the maxima of $F(N)$ as a function of the parameter d/D. As d/D increases, the maximum decreases smoothly. However, there reaches a point (as in figure 4.7(f) where the maximizing value drops suddenly to the lowest value of N. There is thus a "jump" — a phase transition — in the population of agents that can be supported, and a smaller number of agents is more acceptable. For larger values of d/D, only the minimum number of agents is acceptable.

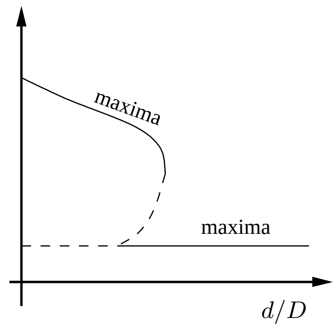


Figure 4.8: Plot of most efficient N as a function of the parameter d/D. A phase transition of sorts of observed.

One point that may be made is that if the agents work closely together, where the “density” is high, then it is more efficient to have more agents. If the problem domain is large enough that the agent density is low, more efficiency is gained by independent operation. (This has interesting implications as applied to intellectual endeavors. People with close interests may work synergistically on a problem, while people with only related interests may get in each others’ way.)

4.8.5 Discussion

It must be conceded that models described above do not, in fact, represent any real system of agents. However, the assumptions in the models are based on practical considerations of the qualitative way that multiagent systems can operate. Up to diffeomorphism, these are feasible models for a system. Furthermore, it known that multiagent systems can experience a “phase transition.” This is typically seen at some sort of boundary between “easy” systems — typically where the number of tasks is much less than the number of agents — and “hard” systems — where the resources of the system are stressed by the demands placed upon it. An ongoing research question is to explore the nature of the phase transition. The simple model presented here sheds light on the nature of the phase transition.

Appendix A

TaskSim Users Manual

A.1 Introduction

The TaskSim simulation was developed by the USU Autonomous Negotiating Teams (ANTs) research group at Utah State University, as part of the USU Analytic Prediction of Emergent Dynamics (APED) project. The simulation will model a limited number of resources as they are assigned to travel to and do work on different jobs. We hope to employ rate equations and praxeid theory along with the simulation to develop an understanding of emergent dynamics in ANT systems.

This manual can be found online at `http://ssl.usu.edu/paul/tasksim/manual/`.

A.2 The TaskSim scenario

In the TaskSim world, there are **jobs** (or **tasks**) that need to be done, and resources that can do them. There is also a **scheduler**, or perhaps series of schedulers, that control one, some, or all of the resources. There is never a way for the schedulers to know exactly when or where new jobs will pop up, but when they do, the schedulers' responsibility is to assign resources to do the jobs. Jobs have certain properties: they have some amount of work that needs to be done, and they have a deadline before which the work should be completed. If a job isn't done on or before its deadline, it ceases to exist, and it is counted as a **failure**. When a job is completed on or before its deadline, it ceases to exist, and it is counted as a **success**. The number of successes and failures are counted during a simulation. Resources might do work on jobs at different rates. The rate at which a resource can work on a certain job is termed the resource's **proficiency** at that job. There may be different **types** of jobs, in which case resources may have different proficiencies for each type. Resources may gain or lose proficiency. Jobs may have **dependencies** on other jobs, meaning that the other jobs must be completed before any work can be done on the job in question. If a job fails, all jobs which are dependent on that job fail as well. Schedulers may or may not take dependencies into account.

A.3 The Simulation

The TaskSim simulation version 2.x models a two-dimensional world. Jobs and resources both have locations in that world, and resources can only do work on a job if they are at the same location. Resources need to move to get to jobs, and they do so at a constant speed. Many resources may occupy the same location. The simulation can use any of several allocation methods. Each method is built into a plugin, or module (a file with a .so extension) so that users may create their own modules. Doing so is beyond the scope of this document. When the simulation is run, the user may select from the available modules. There are three types of jobs in the simulation, called "Green", "Red", and "Blue" jobs. Resources start out with the same proficiency in all job types, but if the "Proficiency Gain" option is enabled, they may become specialized in one or more types. See Simulation options: Resource proficiency gain for more info. Jobs are added to the simulation either by the user (only in the interactive simulation) or at "random". When added at random, the job's work amount and deadline are chosen from configurable uniform distributions. Job locations are chosen uniformly on the world grid. Another quantity, the amount of time before the next random job appears, is chosen from the third configurable distribution. See Simulation options: Random distribution settings for more info. A simulation begins with a certain number of resources, and that number can not change for the life of the simulation. The number of resources with which the simulation starts is configurable.

Running the simulation The TaskSim simulation version 2.x may be run in several ways. When invoked as "armybase", an interactive window is brought up that allows the user to watch the resources moving, watch the jobs being completed, add jobs by clicking on the map, pause and restart the simulation, and so on. This allows the user to learn what the rules of the simulation are, and see what is happening. It is more useful mathematically to run a series of simulations with the same initial parameters to see how an allocation method fares on the average. This is called a "batch simulation" since several are run at once. Batch simulations can be run from the command line (without even any X display) or using the "batch control panel" which lets the user set and change options quite easily. Both of these types of batch simulations are run by invoking "batchsim". The output from a batch simulation is simply the total number of successes and failures, along with a **success percentage** (the number of successes divided by the total number of jobs).

A.3.1 Simulation options

There are a number of options of which a user should be aware. Each can affect the outcome of a simulation. The settings are changed by the user in different ways depending on the style of simulation being run.

A.3.2 Random distribution settings

As stated in the section entitled The simulation, jobs can be added to the simulation at "random". This is always the case when running a batch simulation, since there is no way to add them interactively. When jobs are added in this way, there are several quantities which are selected from uniform distributions. Three of those distributions are configurable.

First: **Deadline**. This refers to the amount of time given to a job to be finished. If the job is not finished before the deadline, it is counted as a failure. The default deadline range is 40-100.

Next, **Work To Do**. This is the amount of work that must be performed on a job before it is completed. The default range is 70-600.

Finally, **Next Job**. This is the amount of time that will elapse before the next random job is added. This information is naturally not made available to the schedulers. Lowering the values means more jobs will be competing for resources, and raising them will ease the workload. The default range is 1-10.

A.3.3 Allocation modules

The method that a scheduler will use to assign resources to jobs is defined by the **allocation module**. After compilation, these modules will appear as files with a .so extension. The user can select from any of the available modules. Those included with this package are:

- **gensched.so** (Generational Scheduling): This method was written specifically for this simulation to be very efficient. Assignment takes into account job and resource locations, job dependencies, and resource proficiencies. Resources give "bids" reflecting how long it would take them to do work on a certain job, including travel time, and the best bids are taken. When there are not enough resources available to accomplish a job, negotiation can occur, and resources can be rescheduled to (or stolen by) the needy job.

- **democratic.so** (Democratic Allocation): When a new job is added, all the resources are split up so that each active job gets an equal number of resources. Location is not taken into account.

- **crisis.so** (Crisis Allocation): When a new job is added, all the resources are redistributed so that jobs get resources in inverse proportion to their time remaining. Location is not taken into account.

- **other.so** (Nothing): No resources are assigned to jobs.

This list does not include several of the allocation types seen in the earlier (Matlab) versions of TaskSim, known as Screaming Generals versions 0.x and 1.x. They may be added in the future.

A.3.4 Resource profi ciency gain

If this option is enabled, resources will gain proficiency when working on a job. They will only gain proficiency in that job type during that time. In essence, they "learn" how to do that type of job in a faster manner. The amount of proficiency gain is not configurable. In the interactive simulation, resource proficiency is indicated by a tint of the appropriate color. A resource that is excellent at Green jobs will appear bright green.

A.3.5 Resource speed

This is the speed a resource will move while traveling, in map units per time unit. This is currently only configurable for batch simulations.

A.3.6 Number of resources

This is the number of resources that will exist in the simulation world. There is no way to change it while a simulation is running. There is no configuration widget to change this value in the interactive simulation, but the number can be changed by editing Scenario files.

A.3.7 Random mode: add random dependencies

If this option is enabled, then when jobs are being added at random, they may also depend on other existing jobs. (A job that depends on other jobs must wait for them to be completed before any work can be done on the job in question). For each preexisting job, there is a constant chance that a new job will depend on it.

A.3.8 Random mode: homogenous jobs

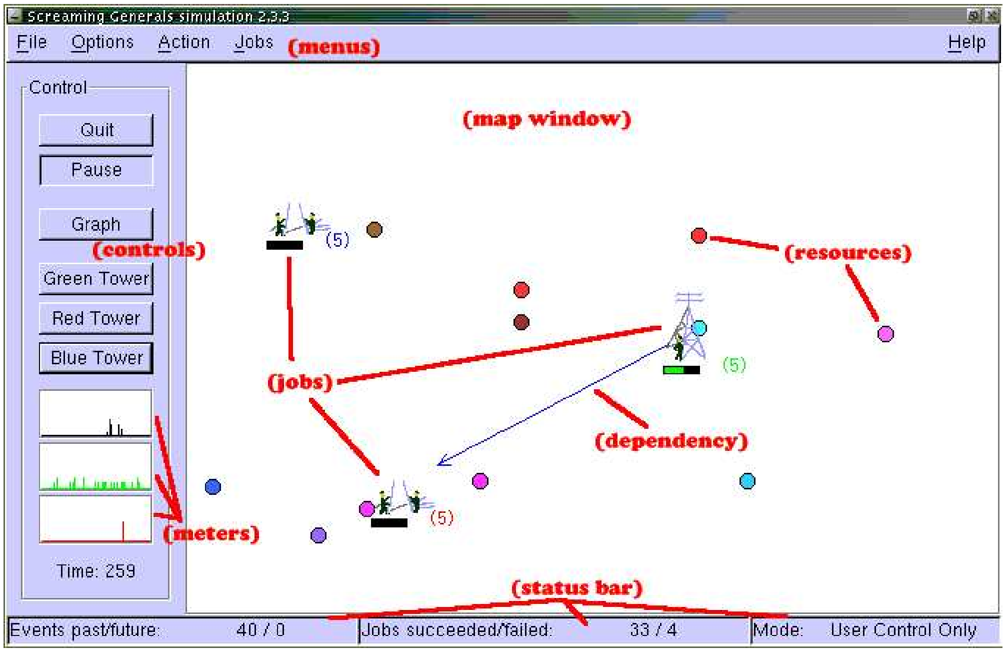
If this option is enabled, all jobs added at random into the simulation will be Green jobs. This removes any consideration of different job types from the simulation.

A.4 armybase - the interactive simulation

The interactive TaskSim simulation is invoked as "armybase". Armybase requires the X Windowing System to run. The simulation will immediately begin after the window pops up. The resources won't do anything, because there are no jobs yet. The Time counter will be moving, though. To stop the simulation at any time, to carefully examine the progress of jobs or make a graph, click on the Pause button. The button will appear depressed while the simulation is paused. To unpause, click the pause button again.

A.4.1 The display

This figure shows the various parts of the interactive user interface.



- The **map window** is the main part of the interface. It shows what is happening in the simulation world.

- **resources** are shown as circles on the map window. They gain a tint of red, green, or blue if they gain proficiency in that type of job. The purple resources, for example, are good at doing Red and Blue jobs.

- **jobs** are shown as radio towers under construction. There is a progress bar under each job, whose length depends on the amount of work a job needs to do. The amount of the progress bar covered in green, red, or blue shows how much of the work has already been done. There is also a number in parentheses under each job. This represents the number of resources that are committed to working on that job now or sometime in the future. The color of the number and on the progress bar shows the type of the job.

- A **dependency** is shown as a blue arrow pointing from one job to another. The job at the arrow's point must wait for the other job to finish before work may begin on it.

- **menus** and **controls** - These allow the user to control what the simulation is doing, add jobs, restart, and so on.

- The **meters** show more information about the successes and failures of jobs. The green (middle) meter shows the number of successfully finishing jobs each time unit. The red (bottom) meter shows the number of failures each time unit. Failures are typically bunched together, especially with dependencies on. The black (top) meter only shows data for some allocation methods. It shows the number of negotations that took place between jobs during each time unit.

- The **status bar** shows the number of jobs that have taken place since time 0 (Events past), the number of jobs scheduled to appear in the future (this is possible with Scenario files), the number of jobs that have succeeded and failed since time 0, and the "Mode". The mode shows whether jobs are currently being added at random.

A.4.2 Adding jobs

To manually add a job, click on one of the Tower buttons in the controls. Then click somewhere on the map window. A job will appear, and resources will (maybe) rush to complete it. Jobs may also be removed manually, although the scheduler won't reschedule anything after they're gone. To do this, right-click on a job and select "Remove".

A.4.3 Adding jobs with dependencies

To add a job that depends on other jobs, you must first select the jobs on which the new job will depend. To select a job, click the middle mouse button on it. If you don't have a middle mouse button, try clicking both buttons at the same time. Your X server may be configured to recognize that combination as a middle click. Selected jobs will show a red box around them. Once you have all the desired jobs selected, use the left button to add a job. It will depend on the selected jobs.

A.4.4 Random job addition mode

You may enter Random job addition mode by selecting "Random Addition Mode" from the Jobs menu. Alternatively, you may simply press R to toggle between Random Addition Mode and User Control Only. When in Random Addition Mode, jobs will appear on the map window as defined by the random distributions (see Setting options below)

A.4.5 Setting options

The options may be found in the Option menu.

- "Allocation Strategy." allows you to select one of the available allocation modules. This may be done at any time, even when the simulation is running.

- "Random Task Parameters" brings up a dialog that allows you to adjust the distributions for Deadline, Work To Do, and Next Job. (see Random distribution settings). The bottom number in the range should be set in the "Min" spin box. The Range slider is not the top value of the range, but the difference between the two values. So, for a Deadline range of 40-100, set Min to 40, and the Range slider to 60. The two checkboxes allow you to enable and disable the Add random dependencies and homogenous jobs options.

- "Proficiency Gain", when enabled, causes resources to gain proficiency when working on a task. See Resource proficiency gain.

- "Show Textbox" attaches a scrolling text box to the bottom of the user interface, which outputs messages whenever a job is added, succeeds, fails, or any one of several other events occurs. It generally displays too much cryptic information to be very useful except in debugging.

A.4.6 Scenario files

Scenario files allow a sequence of added jobs and their dependencies to be saved into a file and replayed again. A scenario file records:

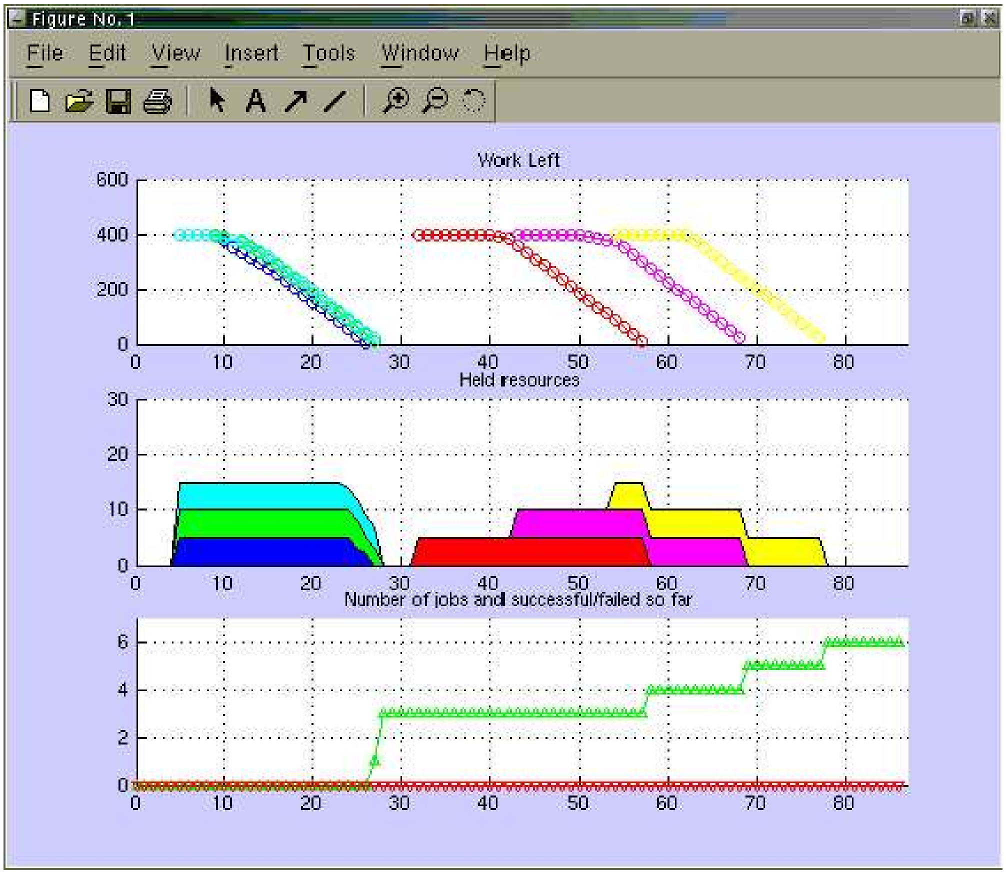
- All jobs that have been added since time 0, by any means, along with their type, starting time, deadline, work to do, location, and dependencies if any. Any jobs that are scheduled to be added in the future are also added to the scenario file.

- Starting locations and beginning proficiency levels for all resources. If you want the simulation to have less resources, make a scenario file and then edit it with a text editor, removing a few of the Resource lines.

To save a scenario file, select "Save Scenario" from the File menu. You will be prompted for a filename. To load a scenario, select "Open Scenario" from the File menu. The time will be reset to 0, and all information about successes, failures, past jobs, future jobs, etc, will be forgotten. The Events future number on the status bar will show a non-zero number if there are any jobs in the scenario file you loaded. If you only want the jobs from the scenario to be added, make sure that Random Addition mode is turned off. To restart the current scenario, choose "Restart" from the Action menu. To forget all past and future job information, and reset the resources, choose "New Scenario" from the File menu. And finally, to forget all past and future job information, while keeping the time and resources as they are, choose "Clear Events" from the Action menu.

A.4.7 Graphing resource interaction

Graphing resource interaction requires that Matlab, version 5 or 6, be installed on the computer. Matlab must also be in the path, so that the TaskSim simulation can start it in the background with the simple command "matlab". If you have matlab, you can graph resource interaction over the last 100 time units by clicking the "Graph" button, or selecting Graph from the Action menu. You may need to wait for some seconds before a Matlab graph appears. The graph will consist of three parts. In the top two sections, each job that existed during the last 100 time units will show up as a certain color. That color will have nothing to do with the job's type. The top section of the graph shows work left against time for all the jobs. The middle section shows what percentage of the available resources were held by each job- again over time. The bottom graph shows a green and a red line. The green line is the number of successes so far, and the red line is the number of failures.



A.5 batchsim - for batch simulations

batchsim is the tool to use for series of non-interactive simulations, or batch simulations. It can run with options on the command line, or bring up a graphical user interface (GUI) to allow the user to set the options visually.

A.5.1 Extra configuration considerations

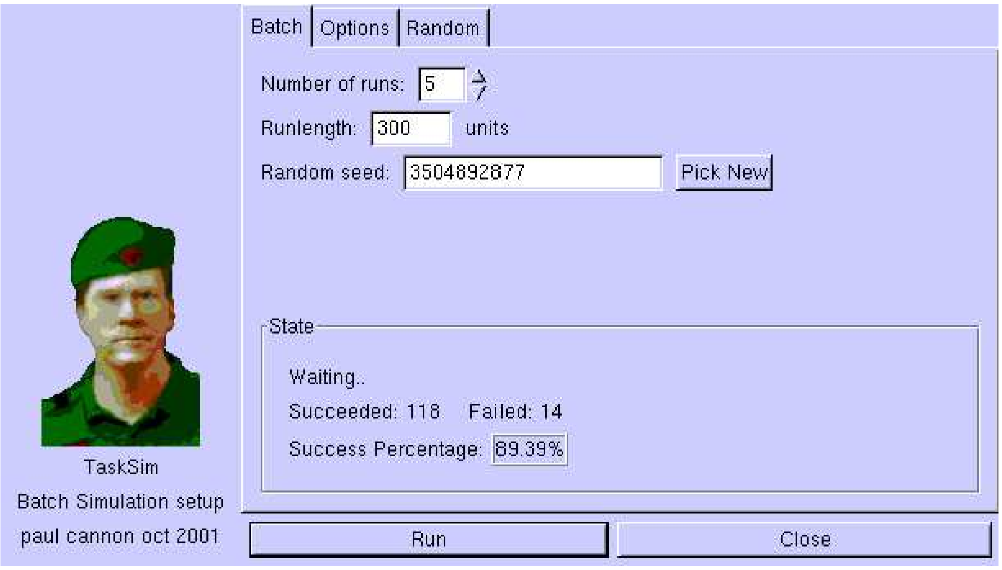
Some configuration parameters need to be set for batch simulations that are not needed in the interactive simulation. These include:

- The **Random seed** - The pseudo-random number generator needs to be "seeded". When a series of simulations is started, the specified seed is planted in the random number generator. This means that if you run the same series of simulations again with the same seed, without changing any other options, the outcome will always be the same. You can write down the seed and use it again later to always get the same results. The simulator can come up with a number for the seed (from the Linux kernel or the system timer) if you do not specify one.

- **Runlength** - the number of time units that each simulation in the series should run.

- **Number of runs** - the number of runs in the series. Each successive run will be different.

A.5.2 The batch control panel



The batch control panel window will come up if you run batchsim without any options, or if you specify a "-g" or "–batch-gui" option among others. It allows you to set the above configuration parameters, along with other options that can be found in the interactive simulation. See Simulation options for more information about them. When you have all the options the way you want them, push the "Run" button. In the "State" box, you can watch the progress of the series. The Run button will remain depressed as long as the series of simulations is running. If it is taking too long, click the depressed Run button again to cancel the run. The results up to that point will be displayed. When you are finished using the batch control panel, click the Close button or close the window in the usual way for your window manager.

A.5.3 The command line interface

batchsim can be run at the command line, and all configuration options are still available as program arguments. You can get a summary of the available options by running ".batchsim -h".

```
$ ./batchsim -h

Simulation options:

--module, -m<filename>: Loads specified allocation module
--timetorun, -t<num>: Runs simulation for <num> time units
--resources, -r<num>: Sets number of resources
--runs, -x<num>: Runs <num> successive simulations- shows average
--dist-deadline, -D<low>-<high>: Sets distribution of job deadline length
```

```
  --dist-jobsize, -R<low>-<high>: Sets distribution of job workload size
  --dist-nextjob, -T<low>-<high>: Sets distribution of time until next job
  --res-speed, -z<num>: Sets resource motion speed
  --use-seed, -s<num>: Sets random seed- useful for duplicating runs
  --dependencies, -d: Turns on random dependencies
  --proficiencies, -p: Turns on resource proficiency gain
  --homogenous, -1: Makes all random jobs the same type
  --verbose, -v: Puts diagnostic output to stderr (noisy)
  --show-info, --show-values, -i: Displays values of many useful parameters
  --batch-gui, -g: Run with the batch simulation control panel (GUI)
  --help, -h: Displays this help text

  When started with no recognized options, the program attempts to open
  the batch simulation control panel.

  $
```

Any of the above options are available when running the batch control panel- simply add a -g or –batch-gui to the options. Since the options are mostly explained above, an exhaustive explanation will not be undertaken here. It is worth noting that the -i option (–show-info, –show-values) is a very valuable one. It shows what random seed is being used, all the random distribution parameters, and the state of nearly all the other options.

To give a feel for the way these options work, here are some example runs:

```
$ ./batchsim -x3 -i
TaskSim Batch Simulator
Version 2.3.7 (against libpaul 0.0.6sg)
brought to you by paul cannon 2001
space software lab/utah state university

Random seed: 870897571
Runlength: 300
Number of runs: 3
Number of resources: 30
Resource move speed: 3.00
Allocation module: gensched.so
Deadline distribution: 40-100
Jobsize distribution: 70-600
Next job distribution: 1-10
Number of random job types: 3
Random dependencies: off
Resource proficiency gain: off
```

```
Run     Succ    Fail    Perc
----    ----    ----    ----
1       51      1       0.98
2       49      1       0.98
3       49      0       1.00
----    ----    ----    ----
Total   149     2       0.99

$ ./batchsim -x3 -i --use-seed=870897571 --module democratic
TaskSim Batch Simulator
Version 2.3.7 (against libpaul 0.0.6sg)
brought to you by paul cannon 2001
space software lab/utah state university

Random seed: 870897571
Runlength: 300
Number of runs: 3
Number of resources: 30
Resource move speed: 3.00
Allocation module: democratic.so
Deadline distribution: 40-100
Jobsize distribution: 70-600
Next job distribution: 1-10
Number of random job types: 3
Random dependencies: off
Resource proficiency gain: off

Run     Succ    Fail    Perc
----    ----    ----    ----
1       6       41      0.13
2       9       36      0.20
3       8       38      0.17
----    ----    ----    ----
Total   23      115     0.17

$ ./batchsim -x=3 --module=crisis -p --res-speed 10
TaskSim Batch Simulator
Version 2.3.7 (against libpaul 0.0.6sg)
brought to you by paul cannon 2001
space software lab/utah state university

Run     Succ    Fail    Perc
```

```
----    ----    ----    ----
1       52      0       1.00
2       58      0       1.00
3       53      0       1.00
----    ----    ----    ----
Total   163     0       1.00

$ ./batchsim --dist-deadline 10-20
TaskSim Batch Simulator
Version 2.3.7 (against libpaul 0.0.6sg)
brought to you by paul cannon 2001
space software lab/utah state university

Run     Succ    Fail    Perc
----    ----    ----    ----
1       14      49      0.22
```

Bibliography

[1] L. Soh and C. Tsatsoulis, "Agent-based argumentative negotiations with case-based reasoning," in *Negotiation Methods for Autonomous Cooperative Systems, Fall Symposium*, AAAI, 2001.

[2] M. Schroeder, L. Boro, and J. McCann, "Efficiency improvements for interactions of web-agents," in *Proc. Fifth Int'l Conf. on Autonomous Agents*, , 2001.

[3] S. Abu-Hakima and F. J. McFarland, C.and Meech, "An agent-based system for email highlighting," in *Negotiation Methods for Autonomous Cooperative Systems, Fall Symposium*, AAAI, 2001.

[4] Y. Gil and S. Ramachandran, "Phosphorus: a taskbased agent matchmaker," in *Proc. Fifth International Conference on Autonomous Agents*, AAAI, 2001.

[5] D. Hart, M. Tudoreanu, and E. Kraemer, "Mobile agents for monitoring distributed systems," in *Proc. Fifth International Conference on Autonomous Agents*, AAAI, 2001.

[6] M. D. Reeves, P. Wellman, and N. Grosof, "Automated negotiation from declarative contract descriptions," in *Proc. Fifth Int'l Conf. on Autonomous Agents*, , 2001.

[7] P. Yolum and P. M. Singh, "Designing and executing protocols using the event calculus," in *Proc. Fifth Int'l Conf. on Autonomous Agents*, , 2001.

[8] H. Jung and M. Tambe, "Towards argumentation as distributed constraint satisfaction, negotiation methods for autonomous cooperative systems," in *Fall Symposium*, AAAI, 2001.

[9] X. Zhang, V. Lesser, and R. Podorozhny, "New results on cooperative, multistep negotiation over a multidimensional utility function," in *Negotiation Methods for Autonomous Cooperative Systems, Fall Symposium*, AAAI, 2001.

[10] M. Frank, A. Bugacov, J. Chen, G. Dakin, P. Szekely, and B. Neches, "The marbles manifesto: a definition and comparison of cooperative negotiation schemes for distributed resource allocation," *Negotiation Methods for Autonomous Cooperative Systems, AAAI Fall Symposium*, 2001.

[11] B. Leake, *Case-Based Reasoning: Experiences, Lessons, and Future Directions*. AAAI Press, 1996.

[12] B. P.J. and B. J., *Constructing Intelligent Agents Using Java*. John Wiley& Sons, 2001.

[13] E. S. Lander and R. V. Lesser, "Sharing metainformation to guide cooperative search among heterogeneous reusable agents," *IEEE Knowledge and Data Engineering*, 1997.

[14] S. Fitzpatrick and L. Meertens, "An experimental assessment of a stochastic, anytime, decentralized, soft colourer for sparse graphs," in *Stochastic Algorithms: Foundations and Applications Proceedings SAGA 2001, Kestrel Institute Technical Report KES.U.01.10* (K. Steinhoefel, ed.), pp. 49–64, Springer-Verlag, Dec. 2001.

[15] C. Liu and J. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, 1973.

[16] R. Armstrong, D. Hensgen, and T. Kidd, "The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions," in *7th IEEE Heterogeneous Computing Workshop (HCW '98)*, pp. 79–87, IEEE, Mar. 1998.

[17] W. C. Stirling, *Satisficing Games and Decisions*. Cambridge (to appear), 2003.

[18] M. A. Goodrich, *On a Theory of Satisficing Control*. Ph.d. dissertation, Brigham Young University, 1996.

[19] I. Levi, *The Enterprise of Knowledge*. MIT Press, 1980.

[20] W. C. Stirling and M. Morf, "A new decision-directed algorithm for nonstationary priors," *IEEE Transactions on Automatic Control*, vol. AC-29, no. 10, pp. 928–930, October 1984.

[21] T. K. Moon and W. C. Stirling, "Satisficing negotiation for resource allocation with disputed resources," in *AAAI Fall Symposium*, 2001.

[22] H. A. Simon, "A behavioral model of rational choice," *Quart. J. Econ.*, vol. 59, pp. 99–118, 1955.

[23] I. Levi, *Decisions and Revisions*. London: Cambridge University Press, 1984.

[24] I. Levi, *The Enterprise of Knowledge*. Cambridge, Massachusetts: MIT Press, 1980.

[25] I. Levi, *Gambling with Truth*. Cambridge, Massachusetts: MIT Press, 1967.

[26] W. C. Stirling, M. A. Goodrich, and D. J. Packard, "Satisficing equilibria: A non-classical approach to games and decisions," *Autonomous Agents and Multi-Agent Systems Journal*, 2001. Forthcoming.

[27] W. C. Stirling, M. A. Goodrich, and R. L. Frost, "Toward a theoretical foundation for multi-agent coordinated decisons," in *Proceedings of The Second International Conference on Multi-Agent Systems*, (Kyoto, Japan), pp. 345–352, 1996.

[28] W. C. Stirling, M. A. Goodrich, and R. L. Frost, "Satisficing intelligent decisions using epistemic utility theory," in *Intelligent Systems: A Semiotic Perspective. Proceedings of The 1996 International Multidisciplinary Conference Conference on Multi-Agent Systems, Volume II: Applied Semiotics*, pp. 268–273, 1996.

[29] W. Stirling, "Multi-agent coordinated decision-making using epistemic utility theory," in *Artificial Social Systems* (E. Castelfranchi and E. Werner, eds.), pp. 164–183, Berlin: Springer-Verlag, 1994.

[30] M. A. Goodrich, W. C. Stirling, and E. R. Boer, "Satisficing revisited," *Minds and Machines*, vol. 10, pp. 79–109, February 2000.

[31] M. A. Goodrich, W. C. Stirling, and R. L. Frost, "A theory of satisficing decision and control," *IEEE Trans. Systems, Man, Cybernet.*, vol. 28, no. 6, pp. 763–779, 1998.

[32] W. C. Stirling, M. A. Goodrich, and R. L. Frost, "Procedurally rational decision-making and control," *IEEE Control Systems Magazine*, vol. 16, pp. 66–75, October 1996.

[33] W. C. Stirling and M. A. Goodrich, "Satisficing games," *Information Sciences*, vol. 114, pp. 255–280, March 1999.

[34] W. C. Stirling and T. K. Moon, "A praxeology for rational negotation," in *AAAI Fall Symposium*, 2001.

[35] R. Axelrod, "http://www.pscs.umich.edu/Software/CC/CC4." Univ. of Michigan Center for the Study of Complex Systems, 1996.

[36] J. Pearl, *Probalistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufmann, 1988.

[37] F. Kschischang, B. Frey, and H. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Info. Theory*, vol. 47, pp. 498–519, Feb 2001.

[38] A. Bergson, "A reformulation of certain aspects of welfare economics," *Quarterly Journal of Economics*, vol. 52, pp. 310–334, 1938.

[39] P. A. Samuelson, *Foundations of Economic Analysis*. Cambridge, MA: Harvard University Press, 1948.

[40] T. W. Sandholm, "Distributed rational decision making," in *Multiagent Systems* (G. Weiss, ed.), ch. 5, pp. 201–258, Cambridge, MA: MIT Press, 1999.

[41] M. Shubik, *Game Theory in the Social Sciences*. Cambridge, Massachusetts: MIT Press, 1982.

[42] R. D. Luce and H. Raiffa, *Games and Decisions*. New York: John Wiley, 1957.

[43] I. Stahl, “An n-person bargaining game in an extensive form,” in *Lecture Notes in Economics and Mathematical Systems No. 141* (R. Henn and O. Moeschlin, eds.), Berlin: Springer-Verlag, 1977.

[44] A. Rubenstein, “Perfect equilibrium in a bargaining model,” *Econometrica*, vol. 50, no. 1, pp. 97–109, 1982.

[45] A. Shaked and J. Sutton, “Involuntary unemployment as a perfect equilibrium in a bargaining model,” *Econometrica*, vol. 52, no. 6, pp. 1351–1364, 1984.

[46] S. Kraus and J. Wilkenfeld, “The function of time in cooperative negotiations,” in *Proceedings of AAAI-91*, pp. 179–184, 1991.

[47] S. Kraus and J. Wilkenfeld, “Negotiations over time in a mutiagent environment,” in *Proceedings of IJCAI-91*, pp. 56–61, 1991.

[48] S. Kraus and J. Wilkenfeld, “A strategic negotiations model with applications to an international crisis,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 32, no. 1, pp. 313–323, 1993.

[49] S. Kraus, J. Wilkenfeld, and G. Zlotkin, “Multiagent negotiation under time constraints,” *Artificial Intelligence*, vol. 75, no. 2, pp. 297–345, 1995.

[50] S. Kraus, “Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents,” *Annals of Mathematics and Artificial Intelligence*, vol. 20, no. 1–4, pp. 111–159, 1996.

[51] S. Russell and E. Wefald, “Principles of metareasoning,” *Artificial Intelligence*, vol. 49, pp. 361–395, 1991.

[52] T. Sandholm and V. Lesser, “Coalitions among computationally bounded agents,” *Artificial Intelligence*, vol. 94, no. 1, pp. 99–137, 1997.

[53] S. Zilberstein, “Satisficing and bounded optimality,” in *Proceedings of the 1998 AAAI Symposium*, pp. 91–94, 1998. March 23–25, Stanford California. Technical Report SS-98-05.

[54] B. E. Kaufman, “A new theory of satisficing,” *The Journal of Behavorial Economics*, vol. 19, no. 1, pp. 35–51, 1990.

[55] M. Oaksford and N. Chater, “A rational analysis of the selection task as optimal data selection,” *Psychological Review*, vol. 101, pp. 608–631, 1994.

[56] M. Wellman, “A market-oriented programming environment and its application to distributed multicommodity flow problems,” *Journal of Artificial Intelligence Research*, vol. 1, pp. 1–23, 1993.

[57] T. Mullen and M. Wellman, "A simple computational market for network information services," in *Proc. of the First International Conference on Multiagent Systems*, pp. 283–289, 1995.

[58] E. Ephrati and J. S. Rosenschein, "Deriving consensus in multiagent systems," *Artificial Intelligence*, vol. 87, no. 1–2, pp. 21–74, 1996.

[59] J. P. Wangermann and R. F. Stengel, "Optimization and coordination of multiagent systems using principled negotiation," *J. Guidance, Control, and Dynamics*, vol. 22, no. 1, pp. 43–50, 1999.

[60] J. Rosenschein and G. Zlotkin, *Rules of Encounter*. Cambridge, MA: MIT Press, 1994.

[61] G. Zlotkin and J. S. Rosenschein, "Mechanisms for automated negotiation in state oriented domains," *Journal of Artificial Intelligence Research*, vol. 5, pp. 163–238, 1996.

[62] G. Zlotkin and J. S. Rosenschein, "Mechanism design for automated negotiation and its application to task oriented domains," *Artificial Intelligence*, vol. 86, no. 2, pp. 195–244, 1996.

[63] G. Zlotkin and J. S. Rosenschein, "Compromise in negotiation: Exploiting worth functions over states," *Artificial Intelligence*, vol. 84, no. 1-2, pp. 151–176, 1996.

[64] P. Cohen and H. Levesque, "Intention is choice with commitment," *Artificial Intelligence*, vol. 42, pp. 262–310, 1990.

[65] P. Cohen, J. Morgan, and M. E. Pollack, eds., *Intentions in Communication*. Cambridge, MA: MIT Press, 1990.

[66] S. Kraus and D. Lehmann, "Knowledge, belief and time," *Theoretical Computer Science*, vol. 58, pp. 155–174, 1999.

[67] M. Lewis and K. Sycara, "Reaching informed agreement in multi-specialist cooperation," *Group Decision and Negotiation*, vol. 2, no. 3, pp. 279–300, 1993.

[68] Y. Shoham, "Agent oriented programming," *Artificial Intelligence*, vol. 60, no. 1, pp. 51–92, 1993.

[69] B. Thomas, Y. Shoham, A. Schwartz, and S. Kraus, "Preliminary thoughts on an agent description lauguage," *International Journal of Intelligent Systems*, vol. 6, no. 5, pp. 497–508, 1991.

[70] M. Wellman and J. Doyle, "Preferential semantics for goals," in *Proceedings of AAAI-91*, pp. 698–703, 1991.

[71] K. Sycara, "Persuasive argumentation in negotiation," *Theory and Decision*, vol. 28, pp. 203–242, 1990.

[72] K. Sycara, “Problem restructuring in negotiation,” *Management Science*, vol. 37, no. 10, pp. 1248–1268, 1991.

[73] S. Kraus, K. Sycara, and A. Evenchik, “Reaching agreements thourh argumentation: a logical model and implementation,” *Artificial Intelligence Journal*, vol. 104, no. 1–2, pp. 1–69, 1998.

[74] D. Zeng and K. Sycara, “Benefits of learning in negotiation,” in *Proceedings of AAAI-97*, pp. 36–41, 1997.

[75] D. Zeng and K. Sycara, “Bayesian learning in negotiation,” in *Int J. Human Computer Systems*, vol. 48, pp. 125–141, 1998.

[76] E. H. Durfee and V. R. Lesser, “Negotiating task decomposition and allocation using partial global planning,” in *Distributed Artificial Intelligence* (L. Gasser and M. N. Huhns, eds.), vol. II, pp. 229–244, London: Pitman/Mortan Kaufmann, 1989.

[77] D. M. Kreps, *Game Theory and Economic Modelling*. Oxford: Clarendon Press, 1990.

[78] M. Slote, *Beyond Optimizing*. Cambridge, MA: Harvard Univ. Press, 1989.

[79] H. A. Simon, “Invariants of human behavior,” *Annu. Rev. Psychol.*, vol. 41, pp. 1–19, 1990.

[80] H. A. Simon, *The Sciences of the Artificial*. Cambridge, MA: MIT Press, third ed., 1996.

[81] D. W. Pearce, *Cost-Benefit Analysis*. London: Macmillan, second ed., 1983.

[82] W. C. Stirling and D. R. Morrell, “Convex Bayes Decision Theory,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 21, pp. 173–183, January/February 1991.

[83] A. K. Sen, *Collective Choice and Social Welfare*. Amsterdam: North-Holland, 1979.

[84] R. Axelrod, *The Complexity of Cooperation*. Princeton, NJ: Princeton Univ. Press, 1997.

[85] J. W. Weibull, *Evolutionary Game Theory*. Cambridge, MA: MIT Press, 1995.

[86] H. Raiffa, *Decision Analysis*. Reading, MA: Addison-Wesley, 1968.

[87] W. Zangwill and C. Garcia, *Pathways to Solutions, fixed points, and equlibria*. Prentice-Hall, 1981.

[88] M. A. Goodrich, *A theory of satisficing control*. PhD thesis, Brigham Young University, 1996.

[89] T. Poston and I. Stewart, *Catastrophe Theory and its Applications*. Boston: Pitman, 1978.

[90] P. Saunders, *An Introduction to Catastrophe Theory*. London: Cambridge University Press, 1980.

[91] V. I. Arnold, *Catastrophe Theory*. Berlin: Springer-Verlag, 2nd ed., 1986.

[92] R. Gilmore, *Catastrophe Theory for Scientists and Engineers*. New York: Dover, 1981.

[93] H. B. Callen, *Thermodynamics and an introduction to Thermostatics*. New York: John Wiley, 1985.

[94] A. Smith, *An inquiry into the nature and causes of the wealth of nations*. London: George Bell, 1896.